

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



ESTUDIO DE COMPORTAMIENTOS EMERGENTES
EN ENJAMBRES DE ROBOTS BASADOS EN MECANISMOS DE
INFLUENCIA

POR

ERICK DE JESÚS ORDAZ RIVAS

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA ELÉCTRICA

JULIO 2016

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



**ESTUDIO DE COMPORTAMIENTOS EMERGENTES
EN ENJAMBRES DE ROBOTS BASADOS EN MECANISMOS DE
INFLUENCIA**

POR

ERICK DE JESÚS ORDAZ RIVAS

**COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA ELÉCTRICA**

JULIO 2016

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
Subdirección de Estudios de Posgrado

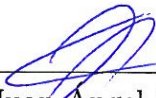
Los miembros del Comité de Tesis recomendamos que la Tesis «Estudio de comportamientos emergentes en enjambres de robots basados en mecanismos de influencia», realizada por el alumno Ing. Erick de Jesús Ordaz Rivas, con número de matrícula 1411496, sea aceptada para su defensa como requisito parcial para obtener el grado de Maestría en Ciencias de la Ingeniería Eléctrica .

El Comité de Tesis



Dr. Luis Martín Torres Treviño

Asesor



Dr. Juan Angel Rodriguez Liñan

Revisor



Dr. Oscar Salvador Salas Peña

Revisor

Vo. Bo.



Dr. Simón Martínez Martínez

Subdirector de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, julio 2016

AGRADECIMIENTOS

Agradezco a mi director de tesis, el Dr. Luis Martín Torres Treviño, por su tiempo, apoyo y orientación en el desarrollo de esta tesis.

Al Dr. Juan Ángel Rodríguez Liñán y el Dr. Oscar Salvador Salas Peña, por sus comentarios y observaciones que ayudaron a mejorar el contenido de esta tesis. A los profesores del Posgrado en Ingeniería Eléctrica, por sus enseñanzas durante mis estudios.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), la Universidad Autónoma de Nuevo León (UANL) y la Facultad de Ingeniería Mecánica y Eléctrica (FIME), por otorgarme la beca y los medios para realizar esta tesis.

A mis compañeros de posgrado, por su apoyo en el transcurso de mis estudios.

A mis amigos, en especial a Gerardo Martínez, por su apoyo, consejos y motivación en el transcurso de mi vida.

A mi familia, por su apoyo constante y motivación para perseverar con mis estudios y, en especial, a mi madre Lic. María Francisca Ordaz Rivas, mi abuela Sra. María de Jesús Rivas Ávila y mi abuelo Sr. Miguel Ordaz Hernández por sus enseñanzas, guía y amor a lo largo de mi vida.

Gracias a Dios, por escuchar mis oraciones, darme fuerza y bendecirme a mí y todas las personas que me rodean.

A mi madre.

*Espero que este nuevo logro en mi vida sea un
motivo por el cual te sientas orgullosa de mí.*

ÍNDICE GENERAL

Agradecimientos	IV
Nomenclaturas	XVII
Resumen	XIX
1. Introducción	1
1.1. Motivación	1
1.2. Antecedentes	3
1.3. Justificación	5
1.4. Hipótesis	6
1.5. Objetivos	6
1.5.1. Objetivo general	6
1.5.2. Objetivos particulares	6
1.6. Metodología	7

1.7. Contribuciones	7
1.8. Contenido de la Tesis	8
2. Robótica de enjambres	9
2.1. Modelo de robots del enjambre	9
2.1.1. Modelo cinemático	10
2.1.2. Modelo dinámico	11
2.2. Arquitectura de los robots	13
2.2.1. Limitaciones sensoriales de los robots	14
2.3. Reglas de comportamiento	16
2.3.1. Implementación de reglas en un sistema embebido	19
2.3.2. Implementación de reglas en una plataforma de simulación	26
3. Descripción experimental	27
3.1. Desarrollo de experimentos físicos	27
3.1.1. Especificaciones de robots en experimentos físicos	28
3.1.2. Sistema de procesamiento de imágenes	29
3.2. Desarrollo de simulaciones	32
3.2.1. Especificaciones de experimentos simulados	33
3.3. Descripción de casos experimentales	33

3.3.1. Primer caso experimental	34
3.3.2. Segundo caso experimental	34
4. Resultados	35
4.1. Resultados de primer caso experimental	35
4.1.1. Experimentos físicos	35
4.1.2. Experimentos por simulación	40
4.2. Resultados de segundo caso experimental	54
4.2.1. Experimentos físicos	54
4.2.2. Experimentos por simulación	65
5. Conclusiones	70
A. Reporte técnico de los robots.	72
A.1. Diseño	72
A.2. Hardware	73
A.2.1. Microcontrolador	73
A.2.2. Sensores y actuadores	73
A.2.3. Alimentación	75
A.3. Software	76

A.4. Funcionamiento	77
 B. Código en Arduino - Reglas de comportamiento	 78
 C. Código en Scilab - Reglas de comportamiento	 89
 D. Código en Scilab - Procesamiento de imágenes	 101

ÍNDICE DE FIGURAS

2.1. Configuración de robot móvil.	10
2.2. Ubicación de sensores y dispositivos en la arquitectura del robot. . . .	14
2.3. Zonas de percepción del robot.	15
2.4. Limitaciones de los sensores de proximidad.	15
2.5. Vectores de dirección de cada sensor.	16
2.6. Zonas de repulsión, orientación y atracción propuestas por Couzin. . .	17
2.7. Nuevas zonas de percepción del robot.	17
2.8. Diagrama de flujo del algoritmo implementado en microcontrolador. .	21
2.9. Diagrama de flujo de las reglas comportamiento del enjambre. . . .	22
2.10. Diagrama de flujo del proceso de repulsión.	23
2.11. Diagrama de flujo del proceso de influencia.	23
2.12. Diagrama de flujo del proceso de atracción.	23
2.13. Diagrama de flujo del proceso de orientación.	24

2.14. Diagrama de flujo del proceso de conversión a sistema de referencia del robot.	25
2.15. Diagrama de estados correspondiente al comportamiento del robot desde el punto de vista del movimiento de los motores.	26
3.1. Etapas de elaboración de experimentos físicos.	27
3.2. Diagrama de flujo de herramienta de procesamiento de imagen.	30
3.3. Sistema de captura para procesamiento de imágenes.	32
3.4. Etapas de elaboración de experimentos por simulación.	32
4.1. Enjambre de robots, sin influencia en segundo 60, con $r_r = 5$ y $r_a = 40$	37
4.2. Enjambre de robots, con influencia en segundo 100, con $r_r = 5$ y $r_a = 40$	37
4.3. Enjambre de robots, sin influencia en segundo 200, con $r_r = 5$ y $r_a = 40$	37
4.4. Enjambre de robots, sin influencia en segundo 60, con $r_r = 10$ y $r_a = 40$	38
4.5. Enjambre de robots, con influencia en segundo 100, con $r_r = 10$ y $r_a = 40$	38
4.6. Enjambre de robots, sin influencia en segundo 200, con $r_r = 10$ y $r_a = 40$	38
4.7. Enjambre de robots, sin influencia en segundo 60, con $r_r = 15$ y $r_a = 40$	39
4.8. Enjambre de robots, con influencia en segundo 100, con $r_r = 15$ y $r_a = 40$	39

4.9. Enjambre de robots, sin influencia en segundo 200, con $r_r = 15$ y $r_a = 40$	39
4.10. Enjambre de robots con $r_r = 5$ y $r_a = 20$, prueba 1	42
4.11. Enjambre de robots con $r_r = 5$ y $r_a = 20$, prueba 2	42
4.12. Enjambre de robots con $r_r = 5$ y $r_a = 20$, prueba 3	42
4.13. Enjambre de robots con $r_r = 10$ y $r_a = 20$, prueba 1	43
4.14. Enjambre de robots con $r_r = 10$ y $r_a = 20$, prueba 2	43
4.15. Enjambre de robots con $r_r = 10$ y $r_a = 20$, prueba 3	43
4.16. Enjambre de robots con $r_r = 15$ y $r_a = 20$, prueba 1	44
4.17. Enjambre de robots con $r_r = 15$ y $r_a = 20$, prueba 2	44
4.18. Enjambre de robots con $r_r = 15$ y $r_a = 20$, prueba 3	44
4.19. Enjambre de robots con $r_r = 5$ y $r_a = 60$, prueba 1	45
4.20. Enjambre de robots con $r_r = 5$ y $r_a = 60$, prueba 2	45
4.21. Enjambre de robots con $r_r = 5$ y $r_a = 60$, prueba 3	45
4.22. Enjambre de robots con $r_r = 10$ y $r_a = 60$, prueba 1	46
4.23. Enjambre de robots con $r_r = 10$ y $r_a = 60$, prueba 2	46
4.24. Enjambre de robots con $r_r = 10$ y $r_a = 60$, prueba 3	46
4.25. Enjambre de robots con $r_r = 15$ y $r_a = 60$, prueba 1	47
4.26. Enjambre de robots con $r_r = 15$ y $r_a = 60$, prueba 2	47

4.27. Enjambre de robots con $r_r = 15$ y $r_a = 60$, prueba 3	47
4.28. Enjambre de robots con $r_r = 5$ y $r_a = 100$, prueba 1	48
4.29. Enjambre de robots con $r_r = 5$ y $r_a = 100$, prueba 2	48
4.30. Enjambre de robots con $r_r = 5$ y $r_a = 100$, prueba 3	48
4.31. Enjambre de robots con $r_r = 10$ y $r_a = 100$, prueba 1	49
4.32. Enjambre de robots con $r_r = 10$ y $r_a = 100$, prueba 2	49
4.33. Enjambre de robots con $r_r = 10$ y $r_a = 100$, prueba 3	49
4.34. Enjambre de robots con $r_r = 15$ y $r_a = 100$, prueba 1	50
4.35. Enjambre de robots con $r_r = 15$ y $r_a = 100$, prueba 2	50
4.36. Enjambre de robots con $r_r = 15$ y $r_a = 100$, prueba 3	50
4.37. Enjambre de robots con $r_r = 5$ y $r_a = 20$	55
4.38. Enjambre de robots con $r_r = 10$ y $r_a = 20$	56
4.39. Enjambre de robots con $r_r = 15$ y $r_a = 20$	57
4.40. Enjambre de robots con $r_r = 5$ y $r_a = 40$	58
4.41. Enjambre de robots con $r_r = 10$ y $r_a = 40$	59
4.42. Enjambre de robots con $r_r = 15$ y $r_a = 40$	60
4.43. Enjambre de robots con $r_r = 5$ y $r_a = 60$	61
4.44. Enjambre de robots con $r_r = 10$ y $r_a = 60$	62

4.45. Enjambre de robots con $r_r = 15$ y $r_a = 60$	63
4.46. Enjambre de robots con $r_r = 5$ y $r_a = 20$	66
4.47. Enjambre de robots con $r_r = 10$ y $r_a = 20$	66
4.48. Enjambre de robots con $r_r = 15$ y $r_a = 20$	66
4.49. Enjambre de robots con $r_r = 5$ y $r_a = 60$	67
4.50. Enjambre de robots con $r_r = 10$ y $r_a = 60$	67
4.51. Enjambre de robots con $r_r = 15$ y $r_a = 60$	67
4.52. Enjambre de robots con $r_r = 5$ y $r_a = 100$	68
4.53. Enjambre de robots con $r_r = 10$ y $r_a = 100$	68
4.54. Enjambre de robots con $r_r = 15$ y $r_a = 100$	68
A.1. Vista explosionada de la estructura del robot	72
A.2. Asignación de pines del sensor ultrasónico HC-SR04	74
A.3. Asignación de pines del sensor infrarrojo TCRT5000.	74
A.4. Módulo convertidor DC-DC y cargador de batería.	75
A.5. Diagrama esquemático	76
A.6. Alimentación del Arduino y los motores	77

ÍNDICE DE TABLAS

2.1. Parámetros del robot.	14
3.1. Alcance de percepción en sensores.	28
3.2. Rango de percepción de zonas de repulsión, orientación, atracción e influencia.	29
3.3. Velocidades en zonas de percepción en experimentos físicos.	29
3.4. Velocidades en zonas de percepción en experimentos simulados.	33
4.1. Resultados de primer caso experimental con robots físicos.	40
4.2. Resultados de primer caso en plataforma de simulación, experimentos del 1 - 3.	51
4.3. Resultados de primer caso en plataforma de simulación, experimentos del 4 - 6.	52
4.4. Resultados de primer caso en plataforma de simulación, experimentos del 7 - 9.	53
4.5. Resultados de segundo caso experimental con robots físicos.	64

4.6. Resultados de segundo caso experimental en plataforma de simulación.	69
A.1. Características de Arduino Micro	73
A.2. Características del sensor ultrasónico	74
A.3. Características de los sensores infrarojos	74
A.4. Características de los motores	75

NOMENCLATURAS

\mathbf{q}_g	Vector de coordenadas generalizadas del móvil.
x	Posición del robot en el eje horizontal.
y	Posición del robot en el eje vertical.
θ	Orientación del móvil.
v	Velocidad linear del móvil.
ω	Velocidad angular del móvil.
ω_r	Velocidad angular de rueda derecha.
ω_l	Velocidad angular de rueda izquierda.
F_d	Fuerza dinámica generada por el móvil.
τ_d	Par dinámico generado por el móvil.
τ_{dr}	Par dinámico generado por motor derecho.
τ_{dl}	Par dinámico generado por motor izquierdo.
m	Masa del móvil.
I_p	Momento de inercia del móvil.
r	Radio de la rueda.
R	Distancia de las ruedas al eje central del móvil.
d	Distancia del centro de masa al eje de las ruedas.
ZOR	Zona de repulsión.
ZOO	Zona de orientación.
ZOA	Zona de atracción.

r_r	Radio de repulsión.
r_o	Radio de orientación.
r_a	Radio de atracción.
Q_k	Zonas de percepción del móvil.
Q_{kr}	Zonas de repulsión del móvil.
Q_{ka}	Zonas de atracción del móvil.
L_r	Zona de percepción de luz derecha.
L_l	Zona de percepción de luz izquierda.
q_{kr}	Valor de sensores en Q_k .
d_r	Dirección deseada de repulsión.
n_r	Vecinos en la zona de repulsión.
r_{ij}	Vector en la dirección del vecino j .
C_m	Centro de masa del enjambre.
δ	Dispersión del enjambre.
M	Total de individuos en el enjambre.
r_i	Vector de posición del individuo i -ésimo.

RESUMEN

Ing. Erick de Jesús Ordaz Rivas.

Candidato para obtener el grado de Maestría en Ciencias de la Ingeniería Eléctrica .

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio: ESTUDIO DE COMPORTAMIENTOS EMERGENTES EN ENJAMBRES
DE ROBOTS BASADOS EN MECANISMOS DE INFLUENCIA.

Número de páginas: 108.

OBJETIVOS Y MÉTODO DE ESTUDIO: La robótica de enjambres es un enfoque reciente para la coordinación de grandes cantidades de robots simples. Las ventajas que ofrece un enjambre de robots son el uso de estructuras físicas simples y económicas, descentralización, escalabilidad y el uso de reglas de comportamiento abstraídas de la conducta de enjambres en la naturaleza, entre otras. La influencia es un factor importante en un enjambre porque permite alterar su comportamiento para realizar una tarea en particular, mientras que la repulsión y atracción también influyen en el comportamiento, aunque más en una formación específica.

En este trabajo, se estudian los comportamientos emergentes en un enjambre de robots simples que dependen de factores de influencia, además de los parámetros

de repulsión y atracción. Se construyeron prototipos de robots móviles y se implementó una plataforma para experimentos, además se desarrolló un algoritmo para la simulación de comportamientos en un enjambre para validar los resultados obtenidos de la experimentación.

CONTRIBUCIONES Y CONCLUSIONES: En los resultados obtenidos, se observa que el factor de influencia afecta de forma descentralizada en el comportamiento del enjambre para la ejecución de una tarea o localización de un objetivo. Este comportamiento emergente cambia al modificar los parámetros de repulsión y atracción. Los resultados demuestran que es posible implementar reglas de comportamiento sencillas en un enjambre de robots simples y cambiar su estructura mediante factores de influencia, sin requerir de información global, métodos centralizados o comunicación entre los robots.

Firma del asesor: _____

Dr. Luis Martín Torres Treviño

CAPÍTULO 1

INTRODUCCIÓN

1.1 MOTIVACIÓN

Un enjambre es un grupo de agentes similares que conviven e interactúan conforme a reglas que rigen su comportamiento de acuerdo a su naturaleza.

Los enjambres de robots son una nueva área de investigación que se inspira en la aplicación de la inteligencia colectiva al diseño y construcción de sistemas multirobot que exhiban características similares a los patrones de comportamiento observados en animales sociales, tales como enjambres de abejas, colonias de hormigas, cardúmenes de peces, parvadas de aves, entre otros [1][2]. Estos seres poseen habilidades individuales muy simples, pero al vivir en grandes grupos logran realizar tareas más eficientemente como la búsqueda y recolección de comida o escapar de algún depredador [3].

De manera opuesta del control centralizado en grandes grupos de robots, el comportamiento del enjambre se basa en el concepto de obtener solamente información local, mediante la interacción de cada individuo con sus vecinos y como consecuencia se espera que emerja un comportamiento colectivo sin tener un líder, un control jerárquico o información global del enjambre [4][5][6][7].

Las principales ventajas de los enjambres de robots son: 1) Flexibilidad para que el enjambre puede adaptarse a un entorno cambiante. 2) Robustez cuando uno o más robots fallan, el grupo puede seguir ejecutando la tarea. 3) Auto-organización, es decir las actividades no son controladas de manera centralizada ni se supervisan localmente. 4) Escalabilidad para adaptarse a cambios en el número de individuos. 5) Económico, el costo total de un enjambre de robots sería inferior al de un solo robot complejo, gracias a la simplicidad de diseño y producción en masa.

La investigación en robótica de enjambres se centra principalmente en cómo los robots interactúan y cooperan para realizar tareas, más que en los detalles de la ejecución de la tarea [8]. Un enjambre de robots puede tener aplicaciones en actividades que involucran grandes cantidades de tiempo y espacio o que pueden resultar peligrosas o inaccesibles para los humanos e incluso para los mismos robots [9][10]. Algunos ejemplos de aplicaciones potenciales se encuentran en exploración minera, búsqueda de sobrevivientes en derrumbes, detección de derrames de petróleo en el mar, monitoreo de recursos, vigilancia, aplicaciones militares e inclusive exploración de otros planetas [11][12].

Los sistemas de robótica de enjambres son concebidos para ser autónomos y tomar decisiones de manera distribuida. En general estas características son consideradas positivas, pero limitan el grado de control sobre el sistema [13]. Se han propuesto distintas alternativas para interactuar con el enjambre y cambiar su comportamiento sin modificar su autonomía y auto-organización [14]. El parámetro de influencia nos permite alterar el comportamiento del enjambre para realizar una tarea en específico, manteniendo las reglas de comportamiento que incluyen los parámetros de repulsión y atracción.

1.2 ANTECEDENTES

Reynolds (1987) [15], propone uno de los primeros modelos de movimiento en grupos de animales. En este modelo, se siguen tres reglas básicas de comportamiento: 1) Evitar colisiones con miembros más cercanos. 2) Tratar de igualar la velocidad de miembros cercanos. 3) Tratar de permanecer cerca de los otros miembros. En este modelo los individuos simulados tienen acceso directo a la posición exacta, orientación y velocidad de todos los objetos a su alrededor, lo cual representa una percepción alejada de las capacidades reales de los animales sociales en la naturaleza.

Iain Couzin et al. (2002) [16], proponen un modelo para simular el comportamiento de individuos moviéndose en grupo, como resultado de tendencias de repulsión, orientación y atracción, basadas en la posición y alineación de los individuos en relación a los demás. Establecen dos reglas de comportamiento: 1) Los individuos tratan de mantener una distancia mínima con otros, en todo momento. 2) Si los individuos no están evadiendo a otros, entonces tienden a atraerse hacia otros individuos y alinearse con sus vecinos.

Las aportaciones de Reynolds y Couzin son la base de gran cantidad de investigación relacionada al comportamiento emergente en enjambres de robots. Sin embargo, hay pocos resultados en la forma de influir o en cambiar el comportamiento del enjambre sin recurrir a un control centralizado.

Se han propuesto distintos tipos de control para influir en los comportamientos emergentes de robots. En el trabajo de Erkment et al. [17], se propone un marco basado en la hidrodinámica de partículas para el control de comportamientos emergentes en enjambres modelados como fluidos. Pimenta et al. [18], utilizaron la misma técnica de hidrodinámica de partículas para crear leyes de control descentralizado, obligando a los robots a comportarse de una manera similar a las partículas del fluido. Por otro lado, Kira et al. [19], presentan dos formas de control, en el primero

utilizan un aprendizaje basado en casos para producir un modelo generalizado de los parámetros y propiedades globales para situaciones específicas. El segundo control utiliza un aprendizaje evolutivo para obtener la colocación y parametrización de los agentes virtuales que influyen en los robots del enjambre. Sin embargo, en la mayoría de los casos, el uso de algún tipo de controlador requiere información global, por lo cual se pierde la descentralización del enjambre.

Bashyal et al. [20], estudiaron el comportamiento de un enjambre y la interacción de un usuario considerando la búsqueda y localización de una fuente de radiación a través de simulaciones. El objetivo es influir en el enjambre sin recurrir al control humano centralizado. Goodrich et al. [21], desarrollaron un control de conmutación que responde a la presencia de un líder o un depredador controlado por un usuario. El uso de un líder guía al colectivo hacia un objetivo, mientras que el depredador lo divide en sub-grupos para distintos enfoques de resolución de problemas. Se destaca que el uso de los líderes descentralizados se puede utilizar para inducir fenómenos emergentes. Estos fenómenos incluyen la capacidad de cambiar de una estructura a otra y volver a la original sin cambiar los parámetros de los agentes. Por otro lado, Shin-Young et al. [22], presentan el uso de mediadores para dar forma a un perímetro de un enjambre. El método del mediador utiliza agentes especiales que operan desde dentro del enjambre. En este enfoque se le permite a un operador humano coordinar múltiples mediadores, evitando la fragmentación y manteniendo la conectividad del enjambre.

Distintos tipos de mecanismos de percepción se han utilizado para influir en el enjambre. Ducatelle et al. [23], estudiaron la navegación cooperativa de enjambres robóticos en el contexto de un escenario de eventos de servicio. Se centran en la cuestión de cómo el enjambre puede informar a sus miembros acerca de los eventos, y guiar a los robots a los lugares de actividades. Proponen una solución basada en comunicaciones inalámbricas con el envío de información de navegación entre ellos para ser guiados a los eventos de servicio. Martínez et al. [24], analizan una alternativa de navegación para enjambres de robots y proponen una estrategia basada

en el movimiento Browniano, en el que cada robot se modela como una partícula cuyo movimiento está influenciado por señales en el medio ambiente. El grado de influencia en el robot corresponde al diseño de la ruta de navegación en el entorno. Pashna et al. [25], propusieron un algoritmo difuso para enjambres de robots, con el fin de rastrear un derrame petrolero simulando que está influenciada por factores ambientales como viento, corrientes y olas. Cada robot comparte su posición y obtiene la de los otros robots en red.

Sartoretti et al. (2014) [26], propusieron una estrategia de control distribuido, permitiendo a individuos converger a lo largo de una curva planar. Los individuos identifican un número de vecinos dentro de un rango de detección y obtienen información local de ellos. La información se procesa para actualizar los parámetros de control y forzar al enjambre converger a lo largo de una curva planar. El marco matemático se basa en ecuaciones diferenciales estocásticas impulsados por un ruido Gaussiano.

1.3 JUSTIFICACIÓN

El estudio de los enjambres de robots es un nuevo enfoque para la coordinación de grandes cantidades de robots simples. Muchos de los modelos propuestos utilizan información global, dependen de la posición exacta de los vecinos o se controlan de forma centralizada. Con el fin de facilitar la implementación física, es necesaria la creación de esquemas que se basen en la información local percibida y no dependan de la posición de los demás robots, es decir, sin el uso de métodos centralizados.

1.4 HIPÓTESIS

El factor de influencia en conjunto con los parámetros de repulsión, orientación y atracción, afecta de forma descentralizada el comportamiento de un enjambre de robots, permitiendo cambiar su formación para realizar tareas específicas.

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

Estudio de comportamientos emergentes en un enjambre de robots que dependen de factores de influencia, además de los parámetros de repulsión y atracción.

1.5.2 OBJETIVOS PARTICULARES

- Realizar el modelado dinámico de un robot con arquitectura diferencial.
- Probar reglas que incluyan repulsión, atracción e influencia. En el caso de la influencia, considerar políticas tipo depredador presa.
- Implementación del factor de influencia en un enjambre de robots considerando fotorresistencias.
- Adquirir y analizar el movimiento de todo el enjambre mediante un sistema de visión.

1.6 METODOLOGÍA

- Llevar a cabo una revisión bibliográfica de los trabajos existentes.
- Desarrollar algoritmos para estudiar el efecto de los parámetros de repulsión y atracción en un enjambre de robots.
- Incluir el factor de influencia en los algoritmos de cada miembro del enjambre de robots.
- Realizar diversos experimentos considerando fuentes de luz para registrar el comportamiento del enjambre de robots considerando factores de influencia positiva.

1.7 CONTRIBUCIONES

En el desarrollo de esta tesis se obtuvieron las siguientes contribuciones:

- Análisis del efecto del factor de influencia en el comportamiento emergente de los enjambres.
- Desarrollo de una plataforma de simulación incluyendo reglas de comportamiento.
- Mejora de un pequeño enjambre de robots para implementar reglas de comportamiento.
- Elaboración de un sistema de visión para el análisis del movimiento del enjambre.

1.8 CONTENIDO DE LA TESIS

El capítulo 2 presenta a los miembros del enjambre, conformado por robots móviles de tipo diferencial, cuyo comportamiento se describe mediante sus modelos cinemático y dinámico. Posteriormente se muestra la arquitectura de cada robot y las limitaciones sensoriales de cada uno. Se explican las reglas de comportamiento y cómo opera cada robot al interactuar con sus vecinos y su entorno.

En el capítulo 3 se describe el desarrollo de los experimentos físicos y simulados. Además, se describen los casos experimentales planteados para el análisis del comportamiento del enjambre. En el capítulo 4, se muestran los resultados obtenidos de los experimentos físicos y simulados.

Por último, el capítulo 5 muestra las conclusiones, así como las propuestas de trabajo futuro.

CAPÍTULO 2

ROBÓTICA DE ENJAMBRES

En este capítulo se describen las características del enjambre, así como las reglas de comportamiento implementadas. Al inicio se propone un modelo de robot móvil con configuración diferencial, en el cual se desarrolla un modelado cinemático y dinámico que es necesario para el análisis del enjambre. Posteriormente se describe la arquitectura de los robots, capacidad sensorial y las limitaciones de cada uno de ellos. Se desarrollan reglas de comportamiento sencillas tomando como base el modelo de Ian Couzin y adaptándolo a robots simples con movimiento en dos dimensiones. Estas reglas son implementadas en cada robot para cambiar su dirección en base a los parámetros de repulsión y atracción y con la interacción de factores de influencia. Por último, se describe el proceso para la validación numérica y experimental de las reglas propuestas.

2.1 MODELO DE ROBOTS DEL ENJAMBRE

Cada miembro del enjambre es representado por un robot móvil con configuración diferencial, debido a su capacidad de maniobra en espacios reducidos (Figura 2.1). Tomando como base el trabajo de A. Bara [27], se obtiene el modelo cinemático y dinámico de esta configuración.

2.1.1 MODELO CINEMÁTICO

En el análisis del modelo de configuración tipo diferencial, el vector de coordenadas generalizadas $\mathbf{q}_g = [x_g, y_g, \theta]^T$, especifica la posición del robot en el marco cartesiano inercial.

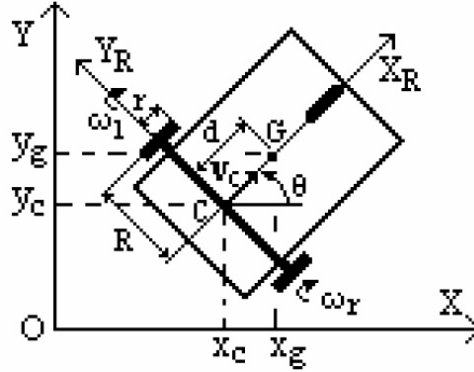


Figura 2.1: Configuración de robot móvil.

En el caso general, las velocidades angulares aplicadas a las ruedas derecha e izquierda son diferentes y por lo tanto el movimiento del robot en el plano horizontal es compuesto. La siguiente relación es válida siempre y cuando el móvil satisfaga las condiciones de rodadura pura.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2R} & -\frac{r}{2R} \end{bmatrix} \cdot \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (2.1)$$

Donde r es el radio de las ruedas, R es la distancia de las ruedas al eje central del móvil, v y ω son las magnitudes de velocidades lineal y angular del punto C del móvil, ω_r y ω_l son las velocidades angulares de las ruedas derecha e izquierda, respectivamente.

Para transformar las velocidades $\mathbf{v} = [v, \omega]^T$, en velocidades cartesianas $\dot{\mathbf{q}}_c = [\dot{x}_c, \dot{y}_c, \dot{\theta}]^T$, se define $\mathbf{S}(\mathbf{q}_c)$ como la matriz Jacobiana.

$$\mathbf{S}(\mathbf{q}_c) = \begin{bmatrix} \cos(\theta) & -d \cdot \sin(\theta) \\ \sin(\theta) & d \cdot \cos(\theta) \\ 0 & 1 \end{bmatrix} \quad (2.2)$$

$$\dot{\mathbf{q}}_c = \mathbf{S}(\mathbf{q}_c) \cdot \mathbf{v} \quad (2.3)$$

Donde d es la distancia del centro de masa al eje de las ruedas.

La cinemática del robot móvil está representada escrita en forma de matriz por:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos(\theta) - \frac{rd}{2R} \sin(\theta) & \frac{r}{2} \cos(\theta) + \frac{rd}{2R} \sin(\theta) \\ \frac{r}{2} \sin(\theta) + \frac{rd}{2R} \cos(\theta) & \frac{r}{2} \sin(\theta) - \frac{rd}{2R} \cos(\theta) \\ \frac{r}{2R} & -\frac{r}{2R} \end{bmatrix} \cdot \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (2.4)$$

2.1.2 MODELO DINÁMICO

La posición del punto gravitacional G se escribe en forma vectorial:

$$\vec{OG} = \vec{OC} + \vec{CG} = \vec{OC} + d \cdot e^{j\theta} \quad (2.5)$$

La aceleración del centro de masa \mathbf{a}_g se obtiene derivando dos veces el vector de posición G .

$$\mathbf{a}_g = (\dot{v} - d \cdot \dot{\theta}^2) \cdot e^{j\theta} + j \cdot (d \cdot \ddot{\theta} + v \cdot \dot{\theta} \cdot e^{j\theta}) \quad (2.6)$$

El primer término es la componente radial que tiene la misma dirección que el vector de desplazamiento y la segunda es la componente tangencial. El movimiento hace adelante en el robot móvil es producido por la fuerza F_d y el movimiento de rotación por el par τ_d , la relación dinámica con estas fuerzas es:

$$\begin{aligned} F_d &= m \cdot \dot{v} - m \cdot d \cdot \dot{\theta}^2 \\ \tau_d &= (I_p + m \cdot d^2) \cdot \ddot{\theta} + m \cdot d \cdot v \cdot \dot{\theta} \end{aligned} \quad (2.7)$$

Por otra parte, estas fuerzas son generadas por pares de torsión de los dos motores, τ_{dr} y τ_{dl} .

$$\begin{aligned} F_d &= \frac{1}{r} \cdot (\tau_{dr} + \tau_{dl}) \\ \tau_d &= \frac{R}{r} \cdot (\tau_{dr} - \tau_{dl}) \end{aligned} \quad (2.8)$$

Tomando estas relaciones (2.7) y (2.8) el modelo dinámico del robot móvil es representado por la forma:

$$\mathbf{M} \cdot \dot{v} + \mathbf{C}(v) = \mathbf{B} \cdot \tau \quad (2.9)$$

Donde

$$\mathbf{M} = \begin{bmatrix} m & 0 \\ 0 & I_p + m \cdot d^2 \end{bmatrix}$$

es la matriz de inercia, simétrica y definida positiva,

$$\mathbf{C}(\mathbf{v}) = \begin{bmatrix} -m \cdot d \cdot \dot{\theta}^2 \\ m \cdot d \cdot v \cdot \dot{\theta} \end{bmatrix}$$

es la matriz de fuerzas centrífugas y de Coriolis,

$$\mathbf{B} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{R}{r} & -\frac{R}{r} \end{bmatrix}$$

es la matriz de entrada y

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_{dr} \\ \tau_{dl} \end{bmatrix}$$

es la matriz de entrada de pares de torsión.

2.2 ARQUITECTURA DE LOS ROBOTS

Los robots propuestos tienen una arquitectura homogénea [28]. Estos robots son de ruedas con configuración diferencial, equipadas con motores de corriente directa. Cada uno cuenta con tres sensores de proximidad de tipo ultrasónico y fotoeléctrico para detectar y medir la distancia hacia otros robots u obstáculos, además de dos fotorresistores que permiten obtener información de la luz percibida en el entorno. Estos sensores están ubicados en el frente y los laterales izquierdo y derecho. Cuentan con un Arduino Micro, en donde se implementan las reglas de comportamiento, y un módulo bluetooth para establecer un ajuste de valores paramétricos en una interfaz usuario-robot. En la Figura 2.2 se ilustra la ubicación de los sensores y dispositivos en el robot. La Tabla 2.1 muestra los valores de los parámetros de los robots propuestos.

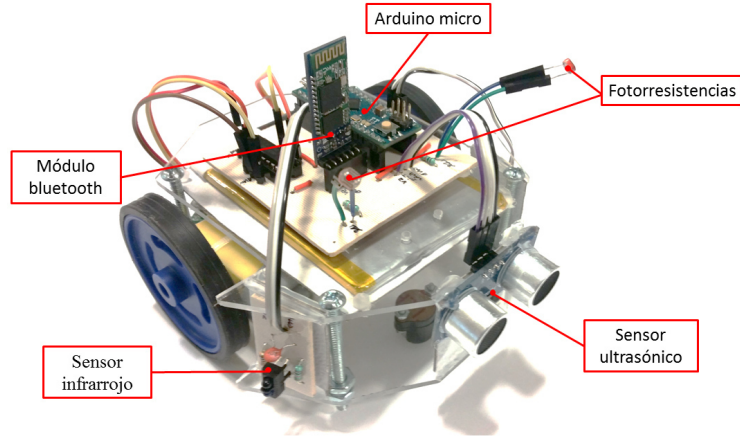


Figura 2.2: Ubicación de sensores y dispositivos en la arquitectura del robot.

Tabla 2.1: Parámetros del robot.

Parámetro		Valor
m	Masa del móvil.	$0.4kg.$
I_p	Momento de inercia.	$0.005kg \cdot m.^2$
r	Radio de la rueda.	$0.03m.$
R	Distancia de las ruedas al eje central del móvil.	$0.05m.$
d	Distancia del centro de masa al eje de las ruedas.	$0.02m.$

2.2.1 LIMITACIONES SENSORIALES DE LOS ROBOTS

Los movimientos permitidos en los robots son giros sobre su propio eje y avance. Los movimientos de reversa están prohibidos para no afectar al momento de evitar obstáculos, ya que en la parte posterior de los robots no hay ningún sensor. Esta limitación se asemeja con las capacidades sensoriales de los seres vivos, que cuentan con un determinado rango de visión y no cubre la totalidad del espacio a su alrededor.

El número de objetos detectados, depende de la cantidad de sensores disponibles en los robots. Sin importar el número de robots que se encuentren dentro de la zona de percepción del sensor, este los considera como uno, ya que no puede determinar la cantidad de vecinos frente a él. La Figura 2.3 muestra las zonas de

percepción por los sensores de cada robot.

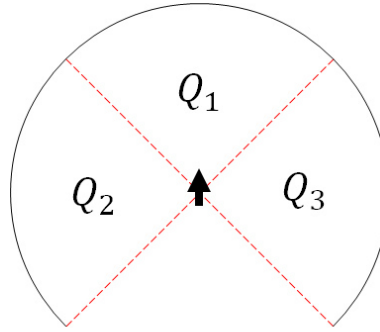


Figura 2.3: Zonas de percepción del robot.

El uso de sensores de proximidad, limita la capacidad sensorial de los robots. Si algún robot se encuentra detrás de los vecinos más cercanos, estos no pueden ser detectados, aunque se encuentren dentro del área de percepción como se muestra en la Figura 2.4.

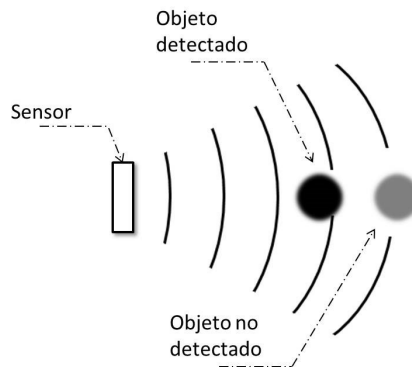


Figura 2.4: Limitaciones de los sensores de proximidad.

Debido a estas limitaciones, los robots sólo cuentan con información sobre la distancia hacia sus vecinos en dirección del sensor de proximidad correspondiente. No se cuenta con información de la orientación ni de la cantidad exacta de vecinos.

2.3 REGLAS DE COMPORTAMIENTO

El cálculo de la dirección deseada para cada miembro del enjambre se basa en el modelo propuesto por Couzin, en donde se suman todos los vectores en la dirección de los vecinos detectados. Debido a las limitaciones sensoriales de los robots se consideran sólo 3 vectores unitarios en la dirección de los 3 sensores con los que cuenta cada robot en un sistema de coordenadas local, como se muestra en la Figura 2.5.

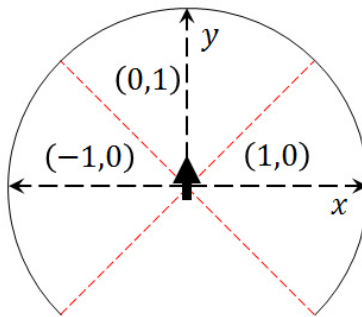


Figura 2.5: Vectores de dirección de cada sensor.

Del modelo de Couzin, se toman zonas de detección locales. La zona de repulsión (*ZOR*) es la más cercana al robot, seguida por la zona de orientación (*ZOO*) y por último la zona de atracción (*ZOA*). El robot se encuentra en el centro de las zonas y el radio de cada zona de detección es denotado por r_r , r_o y r_a respectivamente. La Figura 2.6 ilustra las zonas de detección locales propuestas por Couzin.

Para la implementación en una plataforma de simulación y en pequeños robots que no usan comunicación o mecanismos de detección complejos, el modelo propuesto por Couzin fue simplificado en dos dimensiones. El enjambre solo usa información local de sus sensores para calcular sus movimientos.

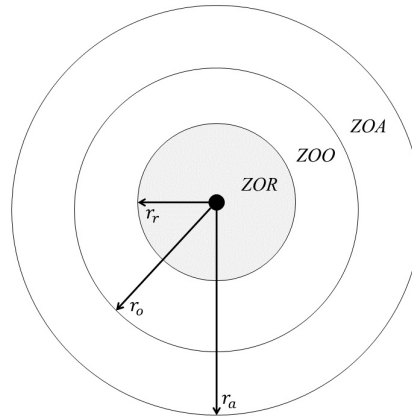
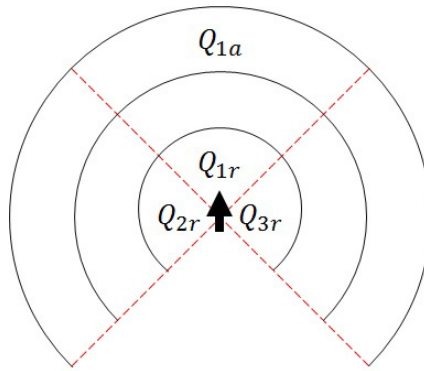
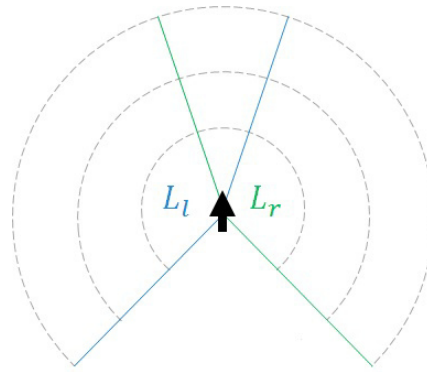


Figura 2.6: Zonas de repulsión, orientación y atracción propuestas por Couzin.

La ubicación de los sensores permite dividir las zonas de comportamiento en nuevas secciones Q_{1r} , Q_{2r} , Q_{3r} y Q_{1a} e incluir dos puntos de recepción de luz L_l y L_r como se muestran en las Figuras 2.7a y 2.7b, respectivamente. La zona Q_1 es la única que permite al robot ser atraído por sus vecinos, mientras que las zonas Q_2 y Q_3 , se limitan solo para repeler a sus vecinos y así evitar colisiones.



(a) Zonas de repulsión y atracción.



(b) Zonas de detección de luz.

Figura 2.7: Nuevas zonas de percepción del robot.

Ya que no es posible conocer la orientación de los vecinos, la influencia toma prioridad en la orientación de los robots. Cuando éstos detecten a sus vecinos dentro de su zona de orientación, la dirección es afectada por la luz percibida en el entorno, guiando a cada uno hacia la fuente de luz percibida con mayor intensidad. Cuando se

detecte una misma intensidad de luz, estos mantienen su dirección para encontrarse con sus vecinos, en la búsqueda por una orientación surgen las reglas de repulsión, atracción e influencia.

Una vez obtenidas las lecturas de los sensores, éstas se procesan para calcular la distancia hacia los vecinos dentro de las zonas de percepción y calcular sus movimientos.

Si un robot detecta a otros en su zona de repulsión, la velocidad disminuye y cambia su trayectoria hacia un vector que apunta en la dirección promedio opuesta a las secciones de repulsión en las que se detectaron sus vecinos, este vector es dado por:

$$d_r = -[(0; 1)q_{1r} + (-1; 0)q_{2r} + (1; 0)q_{3r}] \quad (2.10)$$

Para las secciones Q_{kr} , el valor de los sensores q_{kr} será cero si no se detecta ningún robot en esa sección, o será de uno si se detecta al menos uno.

$$q_{kr} = \begin{cases} 1, & \text{si } n_{Q_{kr}} \geq 1 \\ 0, & \text{si } n_{Q_{kr}} = 0 \end{cases} \quad (2.11)$$

Donde $n_{Q_{kr}}$ es la cantidad de vecinos en la zona Q_{kr} para $k = 1, 2, 3$.

Cuando no se detecta ningún robot en la zona de repulsión, la orientación es priorizada por la influencia. Si se detecta un vecino en la zona de atracción Q_{1a} , surge un aumento de velocidad en la dirección actual del robot.

Cuando no se detecta ningún vecino en las zonas de repulsión y atracción y la intensidad de luz es la misma en L_l y L_r , el robot mantiene su misma dirección hasta encontrarse con otros.

2.3.1 IMPLEMENTACIÓN DE REGLAS EN UN SISTEMA EMBEBIDO

Cada robot del enjambre cuenta con un Arduino Micro, en donde se implementaron las reglas de comportamiento. La programación del microcontrolador se realizó en el entorno de desarrollo integrado de *Arduino* (Arduino IDE 1.6.8). En la Figura 2.8, se muestra un diagrama que describe el Algoritmo 1 implementado en el microcontrolador de forma general. El código en *Arduino* es mostrado en el Apéndice B.

Algoritmo 1: Algoritmo en forma general microcontrolador.

```

Definir constantes y variables;
Iniciar comunicación serial;
repetir
    Leer información desde el puerto serial;
    si  $Val == 1$  a  $9$  entonces
        | Ajustar valor de parámetros;
    fin
    Reglas de comportamiento del enjambre;
    Imprimir datos en el serial;
hasta que detener proceso;

```

DQ_{kr} , para $k = 1, 2, 3$, son variables que indican la distancia en centímetros percibida por los sensores de proximidad frontal, izquierdo y derecho. L_{inf} y L_{sup} , son constantes definidas por el usuario como margen de error en la detección de luz. det_R , det_A y det_O , son variables que indican si existen vecinos u objetos en la zona de repulsión, atracción u orientación, respectivamente, y toman el valor de uno si se detecta al menos un objeto dentro de la zona y cero cuando no hay objetos presentes.

Las reglas de comportamiento del enjambre se encuentran en el diagrama de la Figura 2.9 y en el Algoritmo 2, en donde el factor de influencia se muestra constante.

Algoritmo 2: Reglas de comportamiento del enjambre.

```

repetir
  Obtener datos de los sensores;
  Procesar datos;
  Inicializar variables;
  si  $DQ_k \leq r_r$  entonces // repulsión
    si  $DQ_1 \leq r_r$  entonces
       $q_{1r} \leftarrow 1$ ;
    fin
    si  $DQ_2 \leq r_r$  entonces
       $q_{2r} \leftarrow 1$ ;
    fin
    si  $DQ_3 \leq r_r$  entonces
       $q_{3r} \leftarrow 1$ ;
    fin
     $dd_x \leftarrow -(Q_{1r}[0]q_{1r} + Q_{2r}[0]q_{2r} + Q_{3r}[0]q_{3r})$ ;
     $dd_y \leftarrow -(Q_{1r}[1]q_{1r} + Q_{2r}[1]q_{2r} + Q_{3r}[1]q_{3r})$ ;
     $det_R \leftarrow 1, vel \leftarrow vel_R$ ;
  fin
  si  $det_R \equiv 0$  entonces // influencia
    si  $L_{inf} < L_l \wedge L_{sup} > L_l \wedge L_{inf} < L_r \wedge L_{sup} > L_r$  entonces
      si  $DQ_1 > r_r \wedge DQ_1 \leq r_a$  entonces // atracción
         $dd_x \leftarrow 0, dd_y \leftarrow 1$ ;
         $det_A \leftarrow 1, vel \leftarrow vel_A$ ;
      fin
    en otro caso
      si  $L_l < L_r$  entonces
        TurnRight();
        Stop();
      en otro caso
        TurnLeft();
        Stop();
      fin
    fin
  fin
  si  $det_R \equiv 0 \wedge det_A \equiv 0$  entonces // orientación
    si  $DQ_k \leq r_o$  entonces
       $dd_x \leftarrow 0, dd_y \leftarrow 1$ ;
       $det_O \leftarrow 1, vel \leftarrow vel_O$ ;
    fin
    si  $det_O \equiv 0$  entonces // out of range
       $dd_x \leftarrow 0, dd_y \leftarrow 1$ ;
       $vel \leftarrow vel_O$ ;
    fin
  fin
  Calcular dirección deseada en radianes;
  Conversión al sistema de referencia del robot;
  Escribir datos en actuadores;
hasta que detener proceso;

```

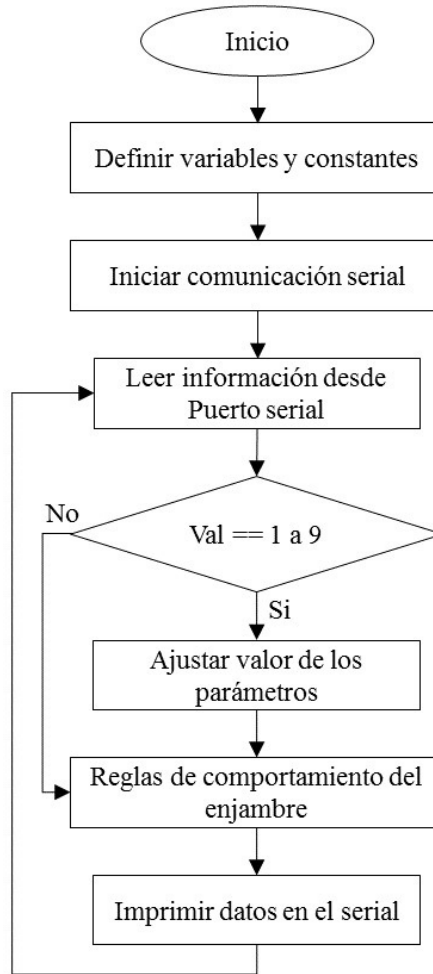


Figura 2.8: Diagrama de flujo del algoritmo implementado en microcontrolador.

Existe una jerarquía en la ejecución de los procesos. La repulsión tiene mayor valor, esto para evitar colisiones con vecinos u objetos, seguida por la influencia, atracción y al final la orientación.

Los procesos de repulsión, influencia, atracción y orientación del diagrama de la Figura 2.9, se detallan en las Figuras 2.10 a 2.13.

Las componentes del vector de la dirección deseada del robot son dd_x y dd_y . Con la expresión $dd_{rad} = atan2(dd_y, dd_x)$ se obtiene el ángulo de orientación en radianes. Para obtener el ángulo de giro dd en grados se aplica la expresión 2.12.

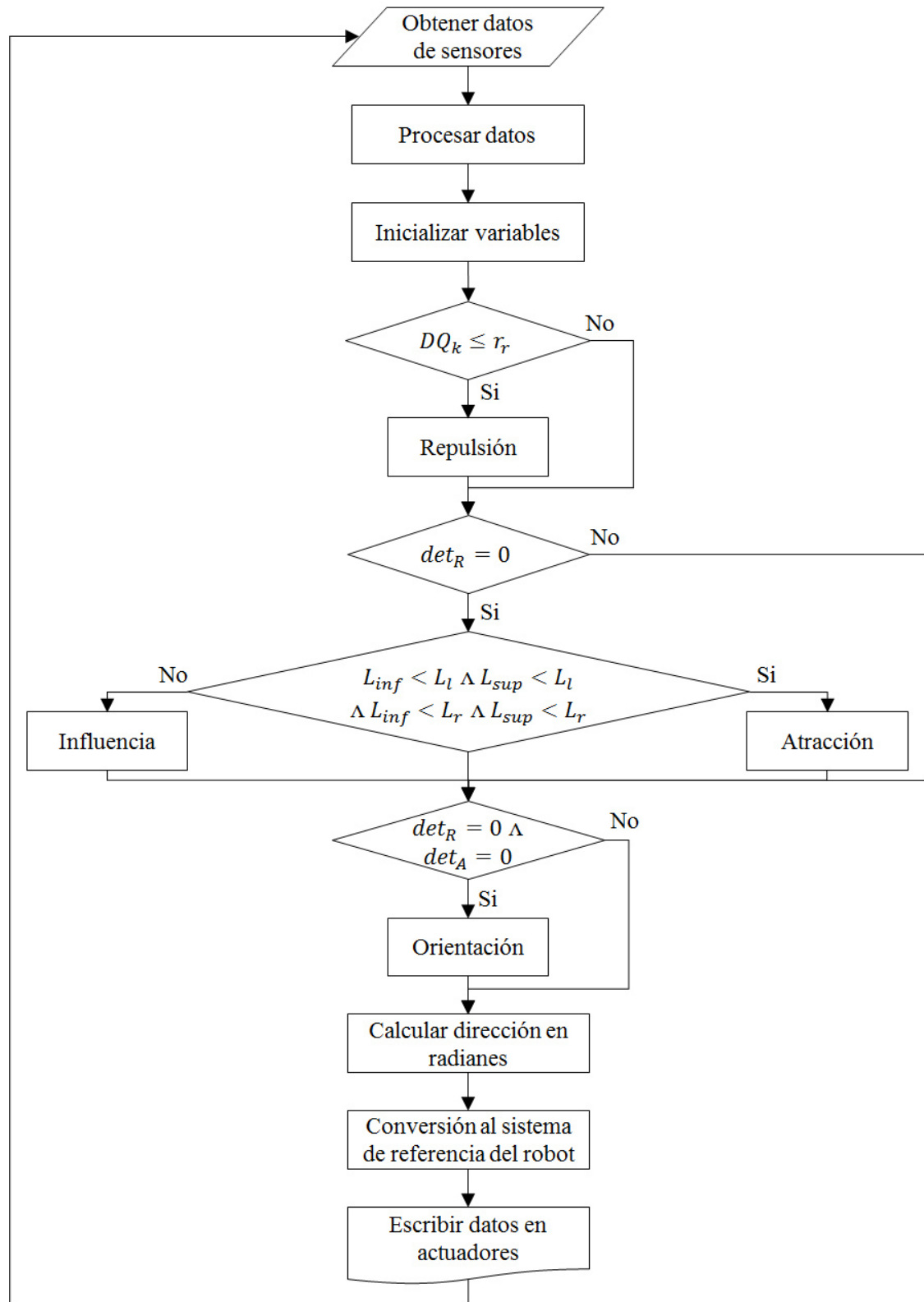


Figura 2.9: Diagrama de flujo de las reglas comportamiento del enjambre.

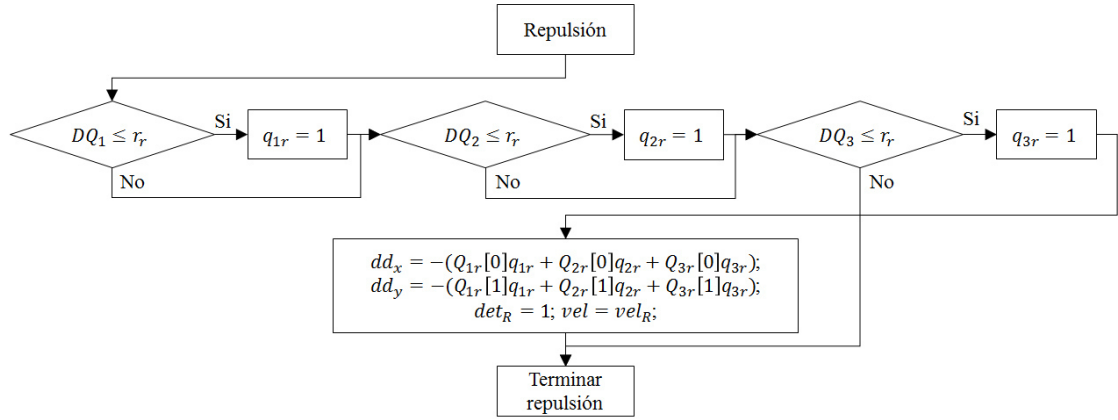


Figura 2.10: Diagrama de flujo del proceso de repulsión.

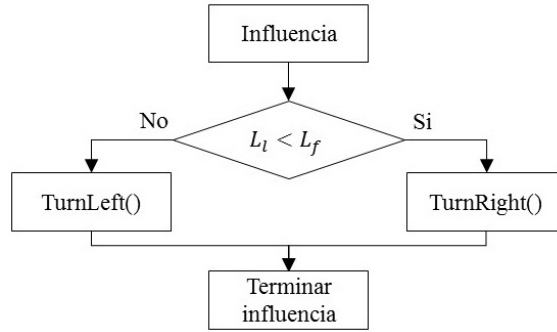


Figura 2.11: Diagrama de flujo del proceso de influencia.

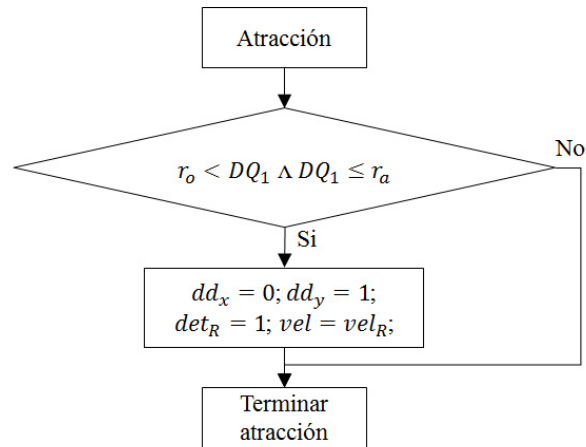


Figura 2.12: Diagrama de flujo del proceso de atracción.

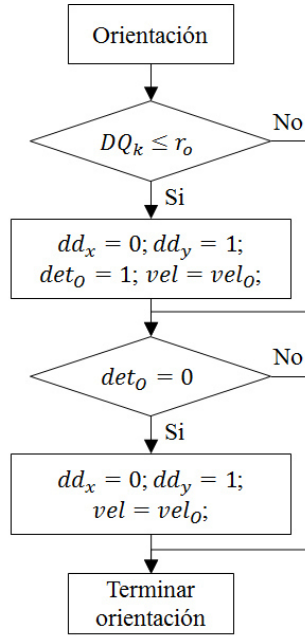


Figura 2.13: Diagrama de flujo del proceso de orientación.

$$dd = dd_{rad} \left(\frac{180}{\pi} \right); \quad (2.12)$$

Para generar el movimiento del robot, en la Figura 2.15 se muestra un diagrama de las funciones utilizadas. En donde ω_r y ω_l corresponden a la velocidad angular del motor izquierdo y derecho respectivamente.

El diagrama de la Figura 2.14 muestra el proceso de la conversión a un sistema de referencia que pueda interpretar el robot para generar los movimientos adecuados según las reglas de comportamiento.

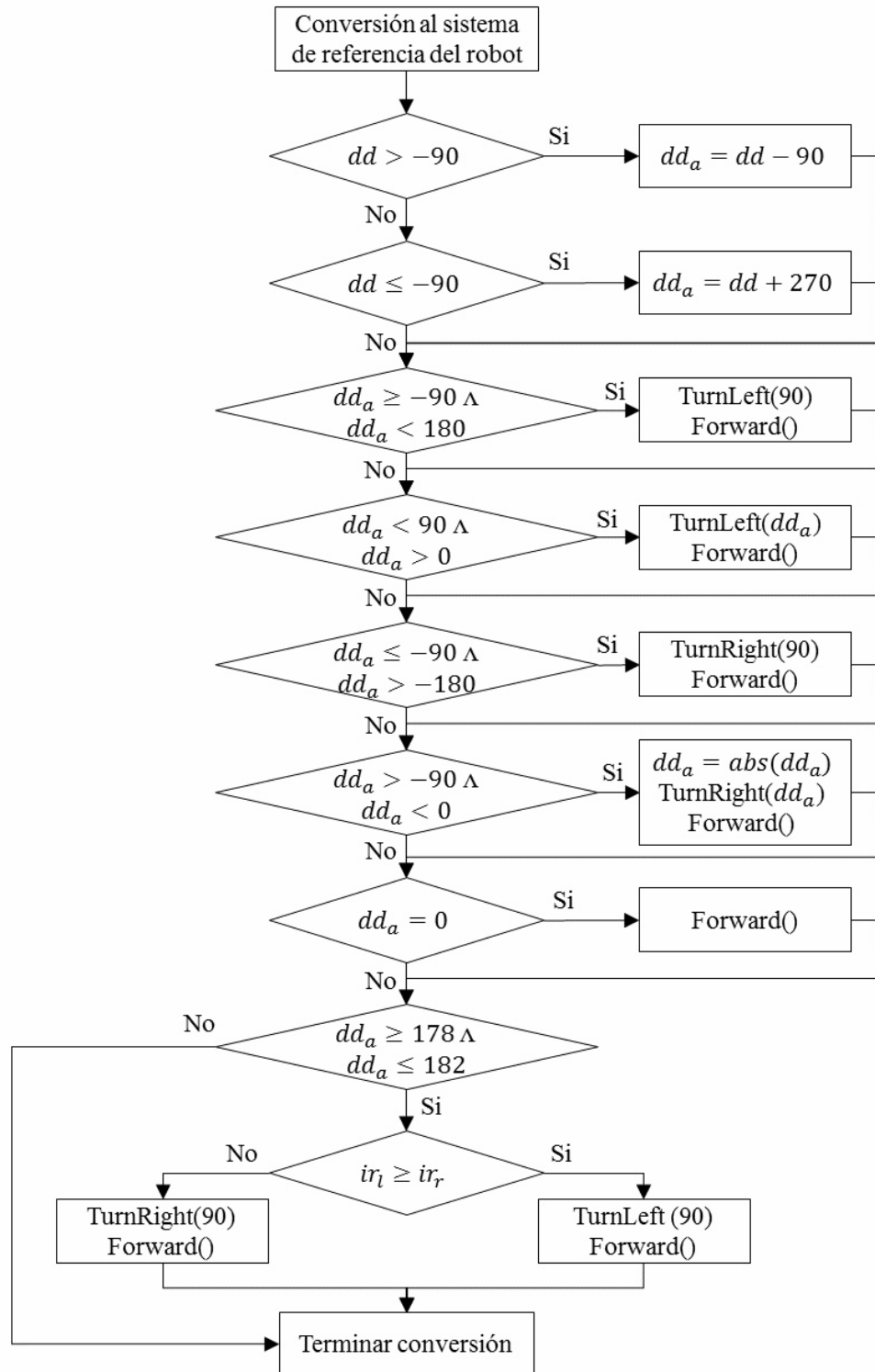


Figura 2.14: Diagrama de flujo del proceso de conversión a sistema de referencia del robot.

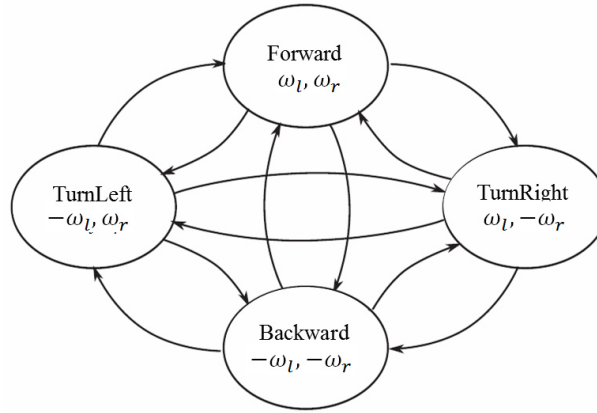


Figura 2.15: Diagrama de estados correspondiente al comportamiento del robot desde el punto de vista del movimiento de los motores.

2.3.2 IMPLEMENTACIÓN DE REGLAS EN UNA PLATAFORMA DE SIMULACIÓN

La finalidad de tener una plataforma de simulación es poder realizar tantos experimentos como sean necesarios, sin importar la cantidad de individuos y el costo de fabricación que estos requieren. Esta plataforma permite explorar los parámetros del modelo propuesto en distintas condiciones, olvidando las limitaciones que se tienen en el sistema físico. Las reglas del enjambre y la capacidad de percepción de cada robot se implementaron en la versión 5.5.2 de *Scilab*, un software de código abierto para análisis numérico, donde se tomó el modelo presentado en la Figura 2.1. El código en *Scilab* es mostrado en el Apéndice C.

CAPÍTULO 3

DESCRIPCIÓN EXPERIMENTAL

En este capítulo se describen las etapas para el desarrollo de los experimentos físicos y simulados. Además del método utilizado para un análisis numérico, mediante una herramienta de cómputo con técnicas de visión. Por último, se exponen dos casos experimentales para el estudio del enjambre. El objetivo de estos experimentos es observar los cambios que surgen en el comportamiento ante variaciones en los parámetros y en la exposición de factores de influencia.

3.1 DESARROLLO DE EXPERIMENTOS FÍSICOS

Las etapas que se siguieron para la obtención de resultados de los experimentos físicos en el estudio del comportamiento de enjambres se muestran en la Figura 3.1.

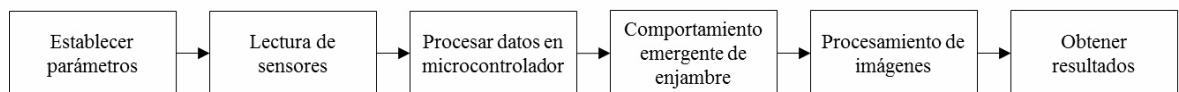


Figura 3.1: Etapas de elaboración de experimentos físicos.

Cada experimento realizado se repite cambiando el valor de los parámetros de repulsión y atracción. Estos se ajustan mediante una interfaz usuario-robot por medio de un dispositivo bluetooth. Los robots obtienen por medio de sus sensores la información percibida en el entorno y esta es procesada por el microcontrolador convirtiéndola en movimiento conforme a las reglas de comportamiento. En el transcurso del experimento, se captura un video y es procesado para obtener datos cuantitativos en el análisis del enjambre.

3.1.1 ESPECIFICACIONES DE ROBOTS EN EXPERIMENTOS FÍSICOS

Se realiza un ajuste en el rango de alcance de los sensores, en la Tabla 3.1 se muestran los detalles.

Tabla 3.1: Alcance de percepción en sensores.

Tipo de sensor	Distancia máxima (cm)	Rango de percepción (grados)
Ultrasónico	100	$\pm 30^\circ$
Fotoeléctrico	20	$\pm 45^\circ$
Fotorresistor	180	$\pm 60^\circ$

Debido a la ubicación y el ajuste realizado en los sensores, el rango de percepción de las zonas de repulsión, orientación, atracción e influencia toman los valores mostrados en la Tabla 3.2. Los radios r_r , r_o y r_a cambian según los parámetros establecidos y el radio r_i es igual a la distancia máxima percibida por los fotorresistores.

Estos valores parten al tomar como origen el vector que apunta en la dirección frontal del robot.

Las velocidades aproximadas, conforme a las zonas de percepción de los robots, se muestran en la Tabla 3.3

Tabla 3.2: Rango de percepción de zonas de repulsión, orientación, atracción e influencia.

Zona	Rango de percepción (grados)
Repulsión	$\pm 120^\circ$
Orientación	$\pm 120^\circ$
Atracción	$\pm 30^\circ$
Influencia	$\pm 140^\circ$

Tabla 3.3: Velocidades en zonas de percepción en experimentos físicos.

Zona	Velocidad (cm/s)
Repulsión	8
Orientación	12.5
Atracción	20
Influencia	20
Fuera de rango	12.5

3.1.2 SISTEMA DE PROCESAMIENTO DE IMÁGENES

Se desarrolló una herramienta con técnicas de visión por computador en *Scilab* para obtener la posición de los miembros del enjambre en un tiempo determinado. En la Figura 3.2 se describe el proceso de esta herramienta con un diagrama de flujo y el código en *Scilab* es mostrado en el Apéndice C.

El video capturado en los experimentos se realiza con un ángulo de inclinación aproximado de 45° con respecto al suelo. Para el procesamiento, se modifica la perspectiva de la imagen a un plano virtual XY con ayuda del software *Adobe Photoshop* versión 2015, un editor de gráficos rasterizados.

Las imágenes son binarizadas al pasar por un umbral y pasan por un proceso que se conoce como de cerrado, en donde se dilata y erosiona la imagen. Este proceso

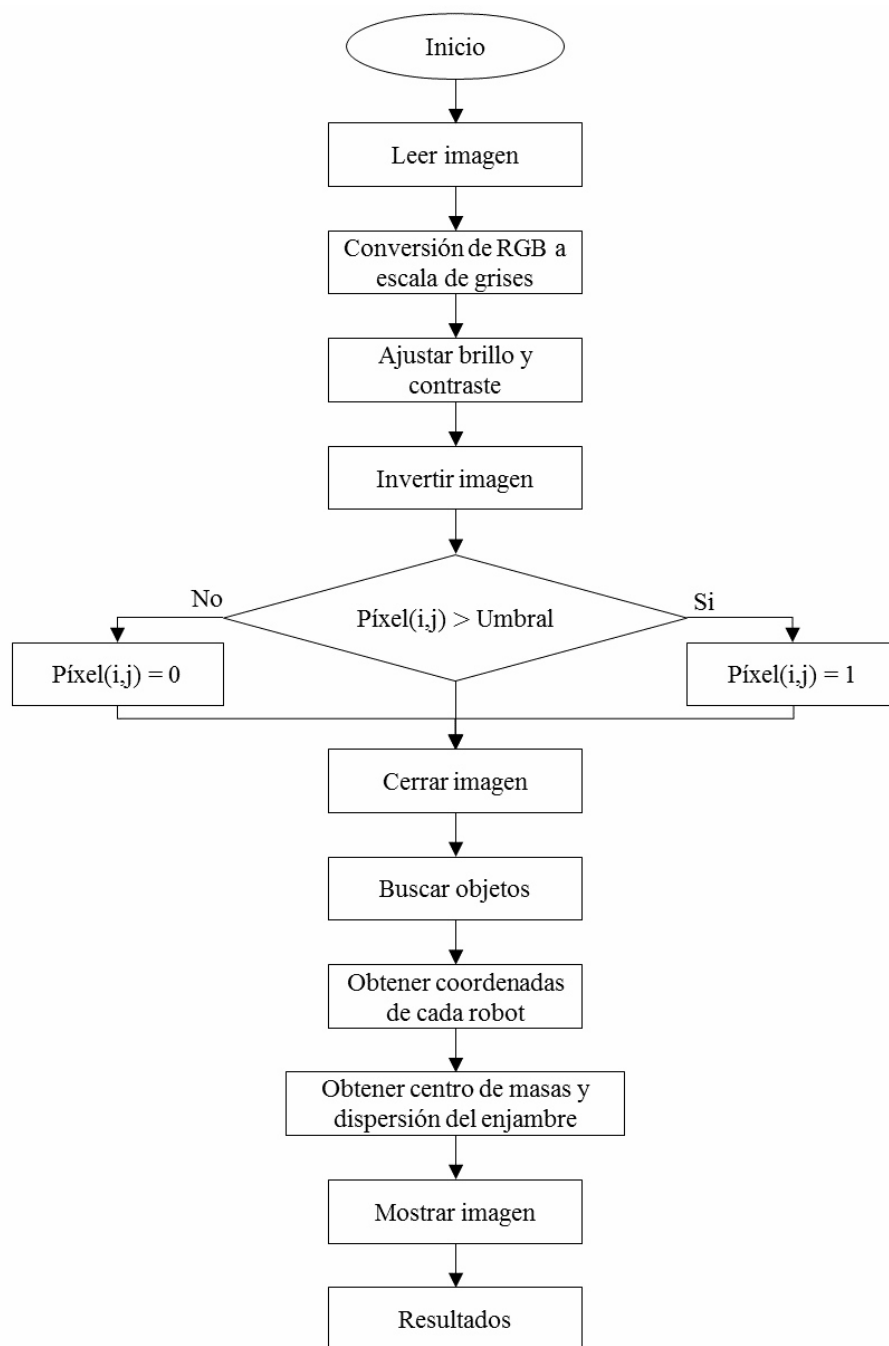


Figura 3.2: Diagrama de flujo de herramienta de procesamiento de imagen.

consiste en marcar con uno los píxeles aislados que se encuentren en cero, es decir, suaviza secciones del contorno, pero generalmente une separaciones estrechas y golfos delgados.

Después del proceso de cerrado, se buscan las regiones conectadas, asignando a cada región un número. Todos los píxeles que no pertenecen a una región son píxeles de fondo y se les da el número cero. Cada región conectada indica un miembro del enjambre. Se obtiene las coordenadas del centro de los robots detectados en un plano virtual XY .

Teniendo las coordenadas xy de cada robot, se obtiene la posición el centro de masa y la dispersión del enjambre aplicando las ecuaciones (3.1) y (3.2).

$$C_m = \frac{1}{M} \sum_{i=1} r_i \quad (3.1)$$

$$\delta = \sqrt{\sum_{i=1} \frac{(r_i - C_m)^2}{M}} \quad (3.2)$$

Donde M es el total de individuos y r_i es el vector de posición del individuo i -ésimo respecto al sistema de referencia supuesto.

El centro de masa y la dispersión son indicadores que nos muestran de manera cuantitativa los cambios en el comportamiento que se genera en el enjambre bajo el factor de influencia. Se forma una elipse en la imagen procesada a partir de estos datos para ser representados de forma visual, en donde se toma el centro de masa y la dispersión como el origen y los semiejes de la elipse, respectivamente. Los semiejes tienen una relación de 1 a 1.3 con respecto a los valores de dispersión.

En la Figura 3.3 se muestra el sistema de captura para el procesamiento de imágenes.

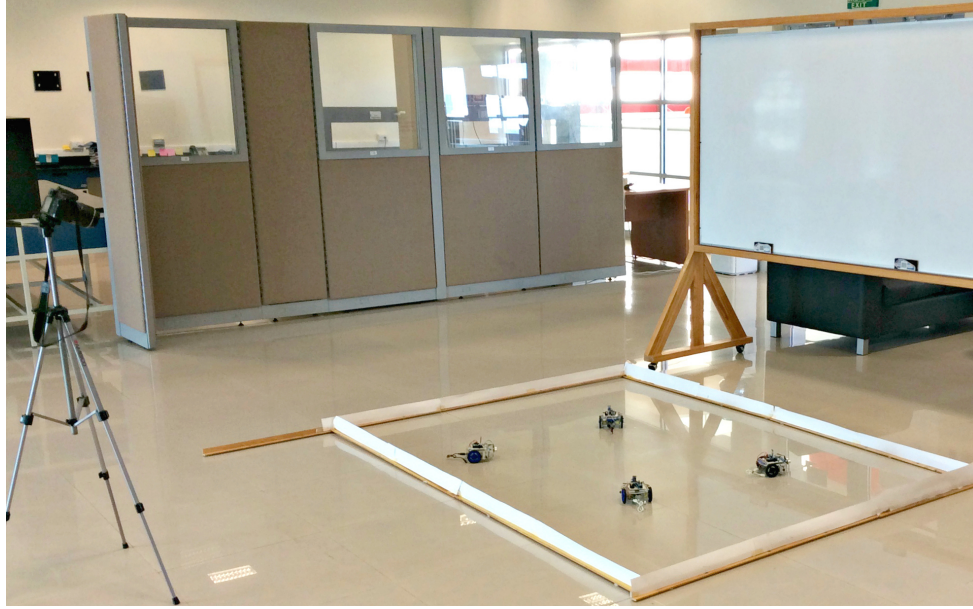


Figura 3.3: Sistema de captura para procesamiento de imágenes.

3.2 DESARROLLO DE SIMULACIONES

Las etapas que se siguieron para la obtención de resultados en la plataforma de simulación se muestran en la Figura 3.4.

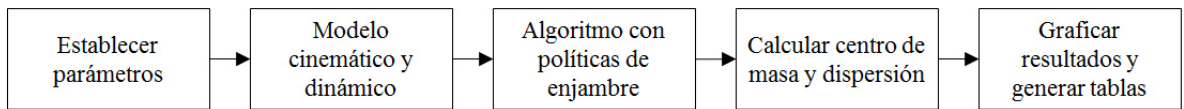


Figura 3.4: Etapas de elaboración de experimentos por simulación.

Los parámetros del enjambre cambian y son ajustados para cada caso experimental. A partir del modelo cinemático y dinámico de las ecuaciones 2.4 y 2.9, se obtienen las condiciones de los robots en la plataforma de simulación, donde se implementaron las reglas de comportamiento. Una vez finalizado el experimento se calcula el centro de masa y dispersión del enjambre, y se generan tablas con los datos recolectados de cada simulación.

3.2.1 ESPECIFICACIONES DE EXPERIMENTOS SIMULADOS

Los robots simulados tienen el mismo rango de percepción que se mostró en la Tabla 3.2. Los radios r_r , r_o y r_a cambian según los parámetros establecidos y el radio r_i es proporcional al lado de mayor tamaño en el área de simulación con una relación de 0.5 a 1, estos valores están dados en decímetros.

Las velocidades de los robots simulados conforme a las zonas de percepción se muestran en la Tabla 3.4

Tabla 3.4: Velocidades en zonas de percepción en experimentos simulados.

Zona	Velocidad (dm/s)
Repulsión	0.5
Orientación	2
Atracción	4
Influencia	4
Fuera de rango	2

En donde las velocidades están dadas en decímetros por segundo. Cada iteración k . corresponde a un intervalo de tiempo de 1s.

3.3 DESCRIPCIÓN DE CASOS EXPERIMENTALES

En esta sección se describen dos casos experimentales en los cuales se influye al enjambre para alterar su comportamiento y formación ante cambios en los parámetros de repulsión y atracción. Para ambos casos se utiliza un factor de influencia positivo y constante.

3.3.1 PRIMER CASO EXPERIMENTAL

En el primer caso, se sitúa el enjambre con una posición aleatoria dentro de un área delimitada. Durante la evolución del experimento se enciende una fuente de luz para estimularlo y atraerlo hacia un punto deseado. Una vez que converge en este punto se apaga la fuente de luz para que este vuelva a su comportamiento original. La finalidad es observar la convivencia de los parámetros de repulsión y atracción ante factores de influencia que se presentan en el entorno y como afectan estos en su comportamiento. Se analizan 3 muestras por experimento: antes, durante y después de aplicar influencia.

3.3.2 SEGUNDO CASO EXPERIMENTAL

En el segundo caso, el enjambre se encuentra concentrado en un punto dentro de un área delimitada. Al inicio del experimento se enciende una fuente de luz en el extremo contrario al enjambre, para que este se traslade hacia el punto influenciado. La finalidad es observar la formación que se adapta en el desplazamiento de punto a punto y los cambios que se presentan ante variaciones en los parámetros de repulsión y atracción. Se analiza una muestra por experimento, donde el enjambre se encuentra en la mitad de recorrido.

CAPÍTULO 4

RESULTADOS

En este capítulo se describen los resultados obtenidos al implementar las reglas de comportamiento en un sistema físico y en una plataforma de simulación. Para cada experimento, se especifican las dimensiones del área de prueba y el tamaño de población del enjambre. Además, se presentan las gráficas y tablas con los resultados resumidos de cada experimento con diferentes valores en los parámetros.

4.1 RESULTADOS DE PRIMER CASO EXPERIMENTAL

4.1.1 EXPERIMENTOS FÍSICOS

El experimento se realiza con 5 robots en un área de prueba de 1.8 x 2.4 m. En el plano virtual, para el procesamiento de imágenes, el tamaño de la imagen es de 500 x 1000 pixeles. En estos experimentos el radio de orientación se mantiene constante con un valor de 15cm. La duración por experimento es de 4 minutos, y se toman tres muestras en los segundos 60, 100 y 200.

La configuración de parámetros de las Figuras 4.1, 4.3 y 4.3, corresponden a una repulsión menor a los demás. Cuando la influencia es activa, el enjambre se reúne en una zona y permanece unido con una dispersión muy baja, provocando que se generen colisiones entre ellos y los muros. Este comportamiento es similar al de las polillas al ser atraídas por la luz, estas vuelan en círculos alrededor y se golpean una y otra vez contra la fuente luminosa hasta morir.

En las Figuras 4.4, 4.5 y 4.6, la repulsión tiene un valor de $10cm$. Cuando la influencia es activa, el enjambre se reúne con una mayor dispersión evitando menos colisiones. En ocasiones los robots al esquivar un obstáculo o a sus vecinos, se colocan en una dirección contraria a la fuente de luz impidiendo que los sensores la detecten y se alejan de ella. Eventualmente regresan al detectarla nuevamente.

En las Figuras 4.7, 4.8 y 4.9, la repulsión es de $15cm$. Cuando la influencia es activa, el enjambre solamente realiza visitas a la zona iluminada, alejándose y regresando a esta. La prioridad del enjambre en esta configuración es evitar colisiones, por lo que llegar a la fuente de luz pasa a ser un objetivo secundario.

En estos experimentos, cuando no hay influencia, el enjambre vuelve a su comportamiento inicial explorando el territorio en búsqueda de una fuente de luz. El radio de repulsión provoca que el enjambre se disperse en el área de prueba manteniéndose más alejado entre sí. Se observa que la dispersión es similar antes y después de influenciar al enjambre.

Cuando la repulsión es corta, los robots tienden a moverse a lo largo de los muros, ya que los sensores no pueden distinguirlos entre otros robots. Al aumentar la repulsión, los robots permanecen más alejados de los muros evitando este comportamiento.

La Tabla 4.1 muestran los datos obtenidos de los experimentos de las Figuras 4.1 - 4.9, marcando con negrita los resultados con influencia.

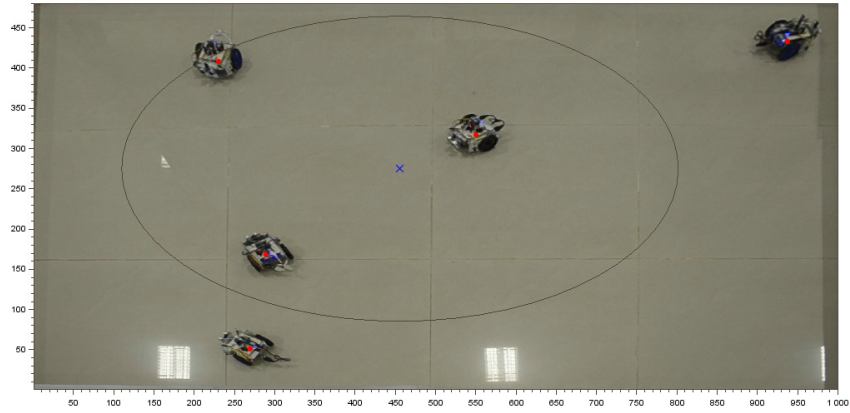


Figura 4.1: Enjambre de robots, sin influencia en segundo 60, con $r_r = 5$ y $r_a = 40$

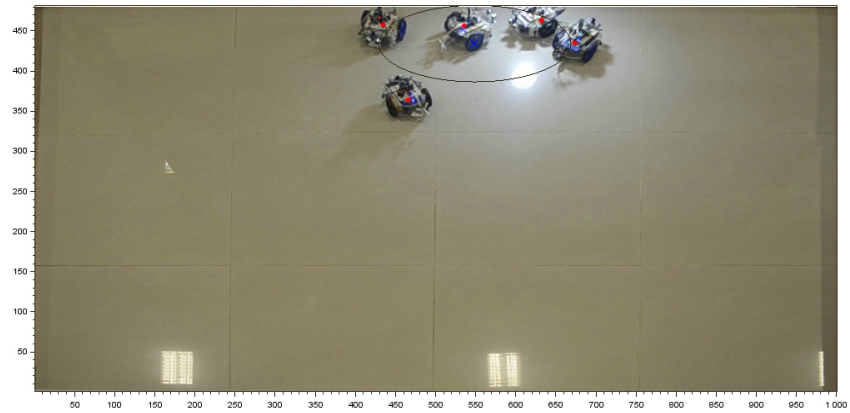


Figura 4.2: Enjambre de robots, con influencia en segundo 100, con $r_r = 5$ y $r_a = 40$

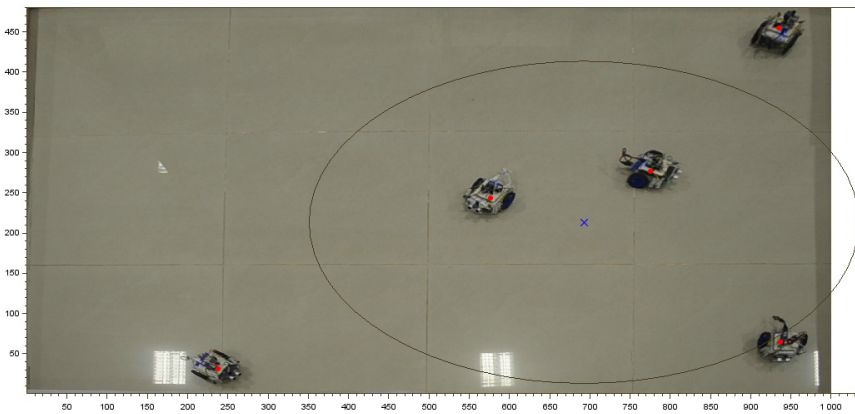


Figura 4.3: Enjambre de robots, sin influencia en segundo 200, con $r_r = 5$ y $r_a = 40$

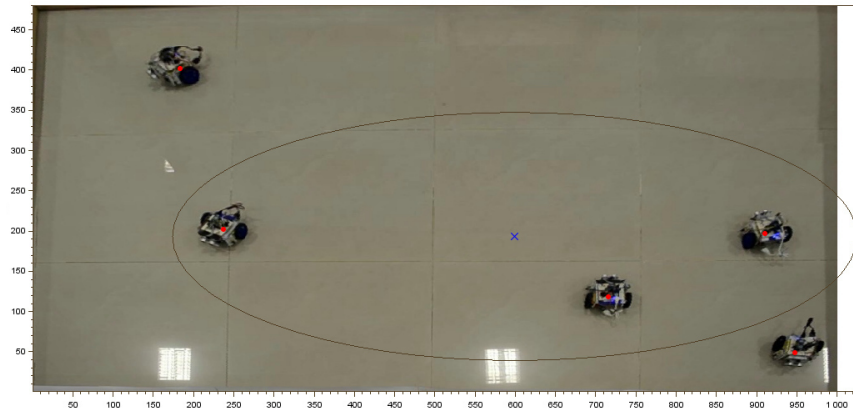


Figura 4.4: Enjambre de robots, sin influencia en segundo 60, con $r_r = 10$ y $r_a = 40$

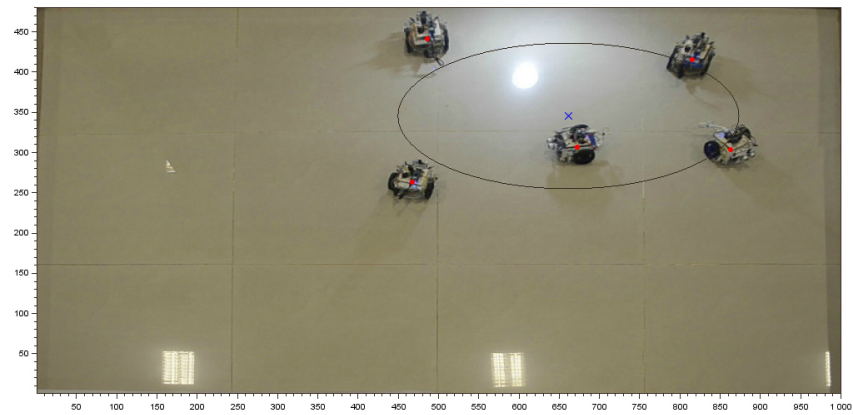


Figura 4.5: Enjambre de robots, con influencia en segundo 100, con $r_r = 10$ y $r_a = 40$

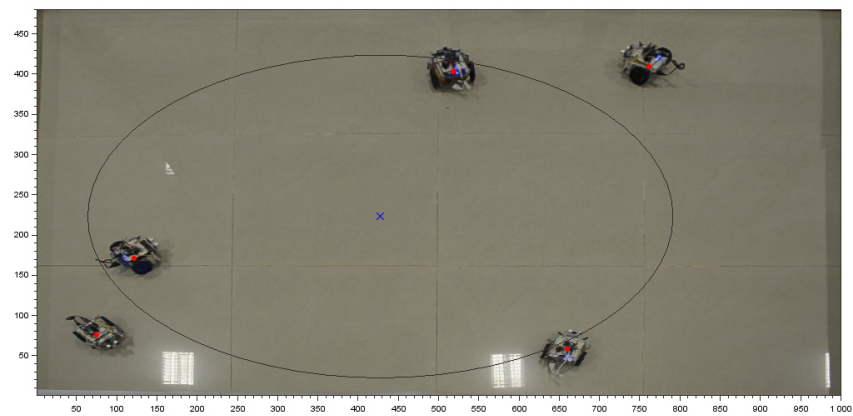


Figura 4.6: Enjambre de robots, sin influencia en segundo 200, con $r_r = 10$ y $r_a = 40$

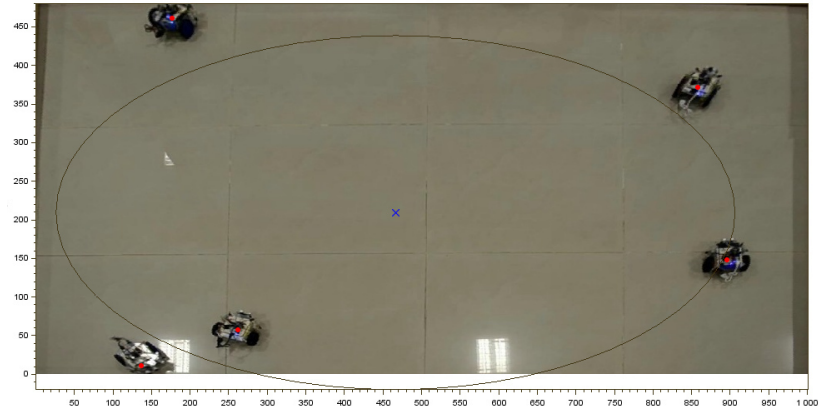


Figura 4.7: Enjambre de robots, sin influencia en segundo 60, con $r_r = 15$ y $r_a = 40$

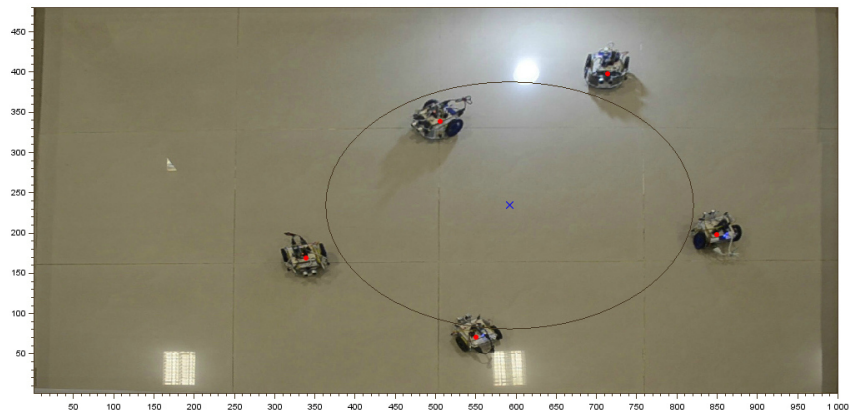


Figura 4.8: Enjambre de robots, con influencia en segundo 100, con $r_r = 15$ y $r_a = 40$

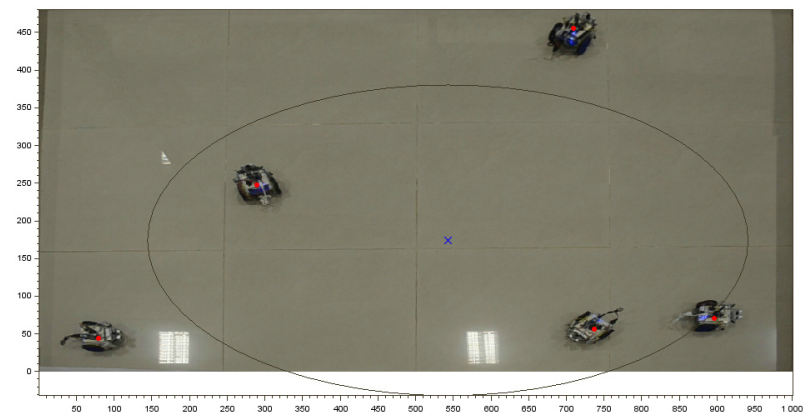


Figura 4.9: Enjambre de robots, sin influencia en segundo 200, con $r_r = 15$ y $r_a = 40$

Tabla 4.1: Resultados de primer caso experimental con robots físicos.

No. Exp	r_r	r_a	T(s)	Cm_x	Cm_y	δ_x	δ_y	Área de elipse
1	5	40	60	455.8	275	266.157	145.819	121927.824
			100	549.1	434.2	92.114	36.668	10611.259
			200	692.8	211.8	262.118	155.693	128207.898
2	10	40	60	704.1	170.8	167.181	78.746	41358.606
			100	661.4	345.6	163.018	69.679	35684.965
			200	427.7	223.1	279.855	154.269	135631.448
3	15	40	60	447	110.5	314.500	57.390	56703.061
			100	592.1	234.4	175.873	118.070	65236.148
			200	542.8	174.2	306.502	158.487	152607.916

4.1.2 EXPERIMENTOS POR SIMULACIÓN

Las simulaciones se realizaron con 25 individuos en un área de prueba de 24 x 24 m. Las gráficas presentadas se muestran en unidad de decímetros. En estos experimentos el radio de orientación se mantiene constante con un valor de 15dm. Se simula una fuente de luz colocándola en las coordenadas [216,24] del área de prueba. Las simulaciones se repitieron 3 veces para cada configuración paramétrica. Por cada simulación se hace un total de 300 iteraciones, y se toman tres muestras en la iteración 100, 180 y 300.

Los resultados presentados validan los experimentos físicos al observar el mismo comportamiento cuando el valor de repulsión cambia. Las Figuras 4.10 - 4.12 muestran los resultados con menor dispersión y con una concentración del centro de masa más cercana a la zona de influencia. En las Figuras 4.13 - 4.15, se presenta un aumento en la dispersión, al igual que en la Figuras 4.16 - 4.18, donde esta es mayor. Al tener un radio de atracción bajo, los individuos buscan la zona de influencia sin importar estar dentro de un grupo o permanecer unido a el enjambre. Una vez reunido y desactivada la zona de influencia este vuelve a su distribución original.

Al aumentar el radio de atracción a un valor medio, los individuos se atraen unos con otros formando grupos más grandes, esto se presenta con mayor frecuencia cuando el radio de repulsión es bajo, Figuras 4.19 - 4.18. Con un radio de repulsión medio y alto, como en las Figuras 4.22 - 4.24 y 4.25 - 4.26, respectivamente, la dispersión aumenta.

En ocasiones, cuando la influencia es activa, los individuos prefieren permanecer unidos a los grupos formados en el enjambre, hasta que alguno de ellos toma la decisión de moverse hacia la zona de influencia y es seguido por los demás integrantes del grupo. Grupos de mayor tamaño localizan más rápidamente la zona de influencia, al haber una mayor densidad de buscadores. Este comportamiento es similar a los cardúmenes de peces en búsqueda de alimento, donde el pez puede aprender a anticipar su momento de alimentación como producto de diferentes estímulos, entre ellos la aglomeración del cardumen cerca de la zona de alimentación.

Con un radio de atracción alto, los individuos convergen con mayor rapidez a la zona de influencia, ver Figuras 4.28 - 4.30. Los grupos con mayor densidad permanecen constantes, mientras los de menor tienden a desaparecer ya que los individuos abandonan estos con mayor facilidad al ser atraídos por grupos más grandes. A mayor radio de repulsión, se forman grupos con mayor dificultad, ver Figuras 4.31 - 4.33 y 4.34 - 4.36.

Los resultados de las simulaciones con radio de atracción igual a 20, 60 y 100dm. se muestran en las Tablas 4.2, 4.3 y 4.4, respectivamente. En las tablas se marca con negrita los resultados con influencia. También se marca el promedio obtenido de las 3 pruebas realizadas por experimento de cada configuración paramétrica.

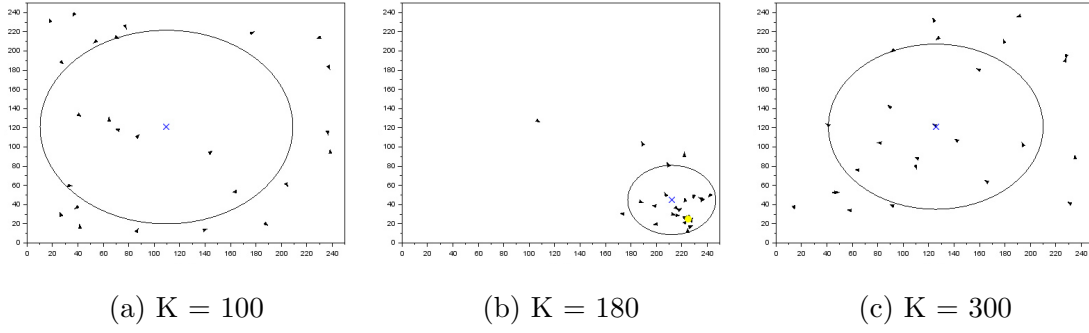


Figura 4.10: Enjambre de robots con $r_r = 5$ y $r_a = 20$, prueba 1

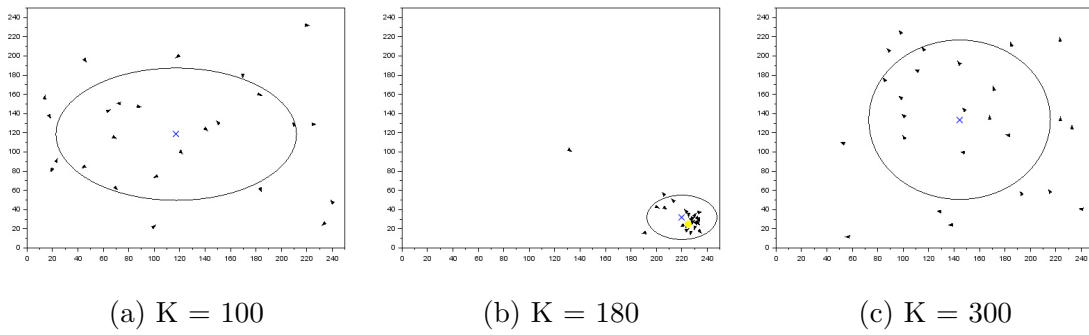


Figura 4.11: Enjambre de robots con $r_r = 5$ y $r_a = 20$, prueba 2

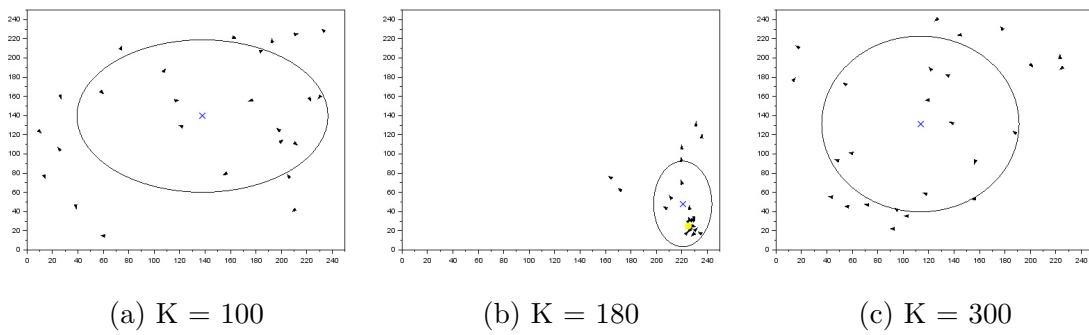


Figura 4.12: Enjambre de robots con $r_r = 5$ y $r_a = 20$, prueba 3

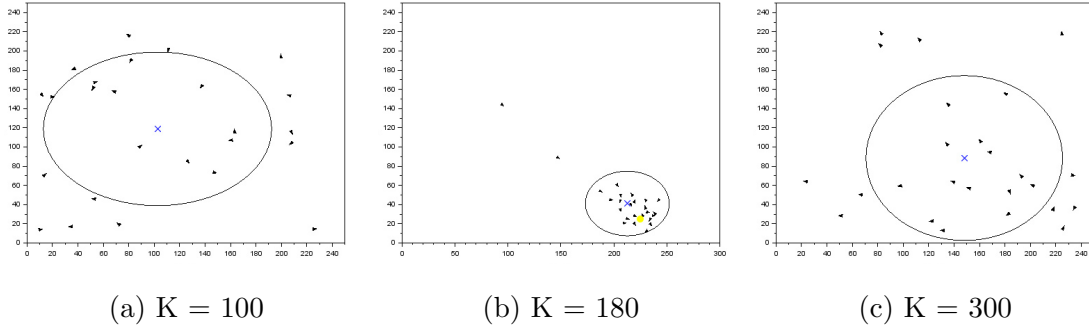


Figura 4.13: Enjambre de robots con $r_r = 10$ y $r_a = 20$, prueba 1

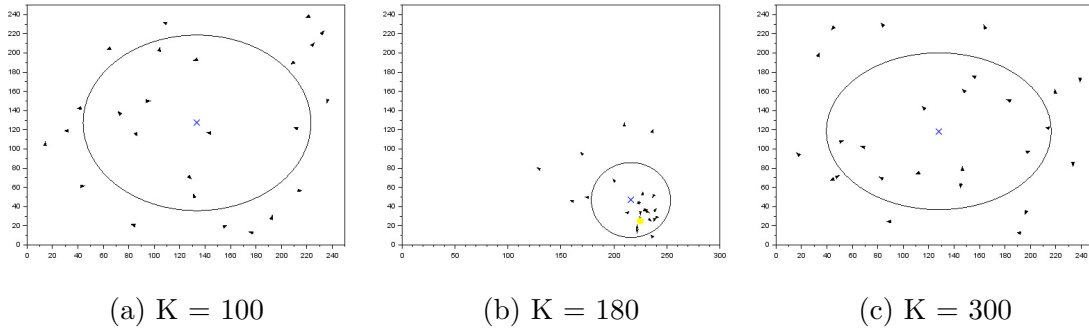


Figura 4.14: Enjambre de robots con $r_r = 10$ y $r_a = 20$, prueba 2

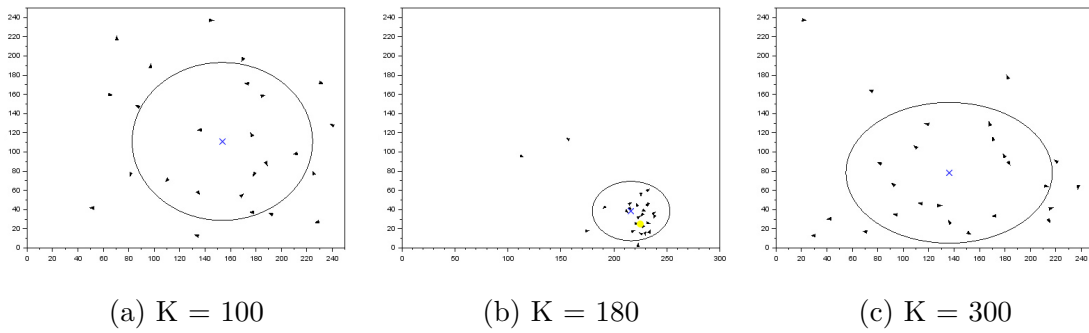
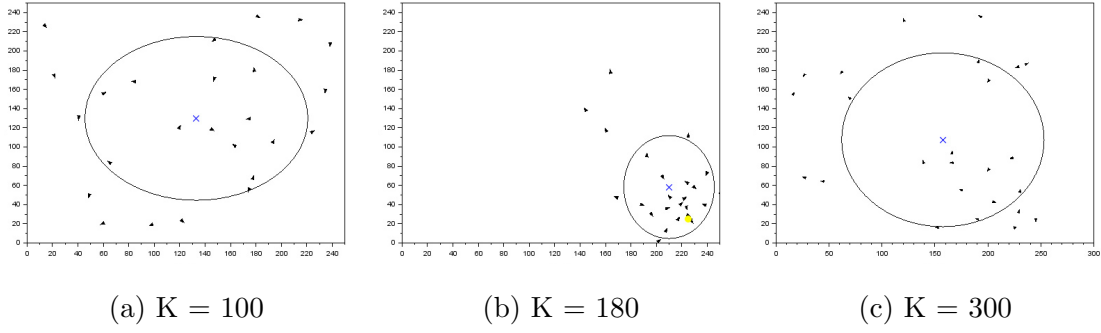
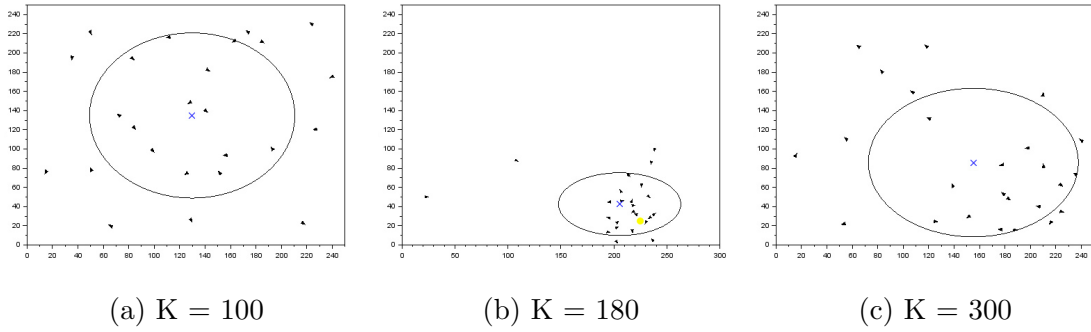
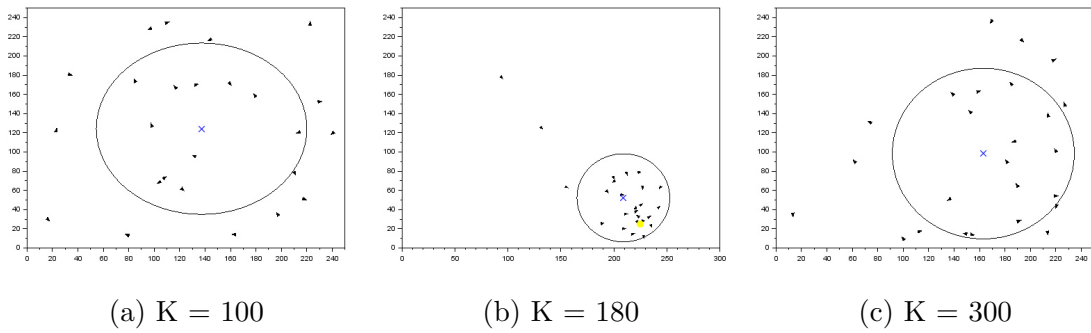


Figura 4.15: Enjambre de robots con $r_r = 10$ y $r_a = 20$, prueba 3

Figura 4.16: Enjambre de robots con $r_r = 15$ y $r_a = 20$, prueba 1Figura 4.17: Enjambre de robots con $r_r = 15$ y $r_a = 20$, prueba 2Figura 4.18: Enjambre de robots con $r_r = 15$ y $r_a = 20$, prueba 3

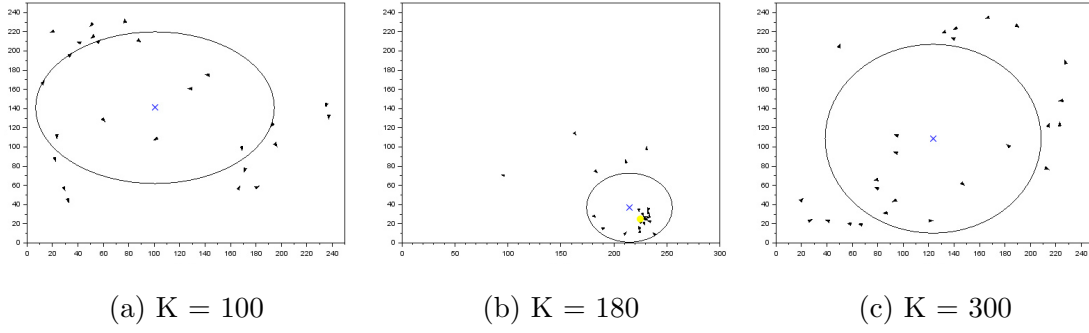


Figura 4.19: Enjambre de robots con $r_r = 5$ y $r_a = 60$, prueba 1

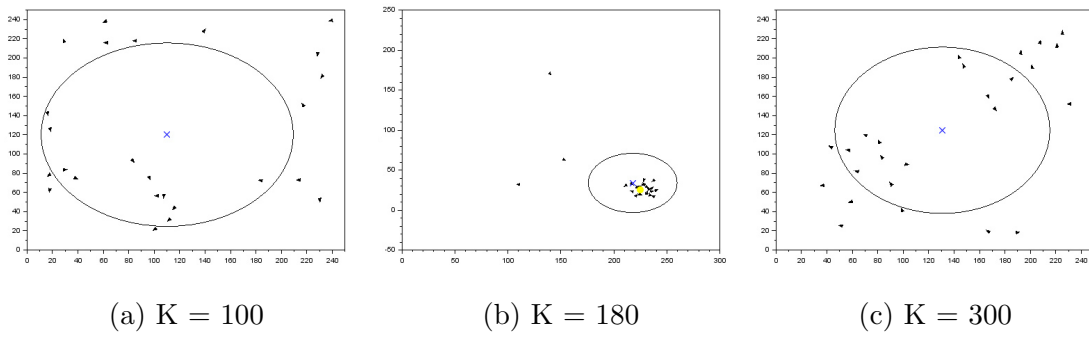


Figura 4.20: Enjambre de robots con $r_r = 5$ y $r_a = 60$, prueba 2

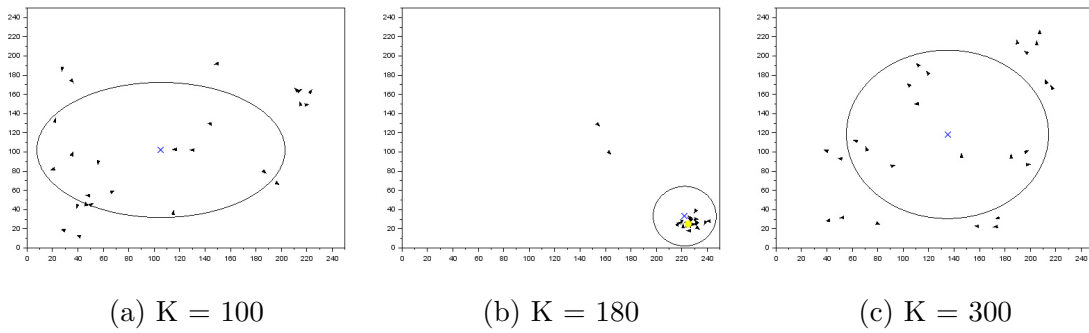
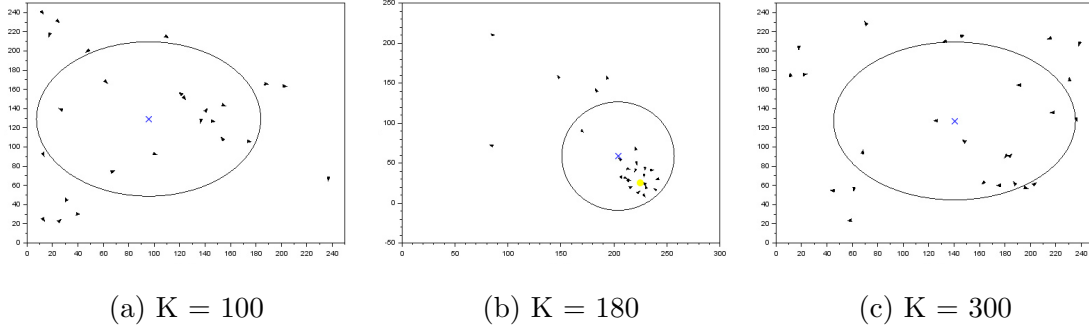
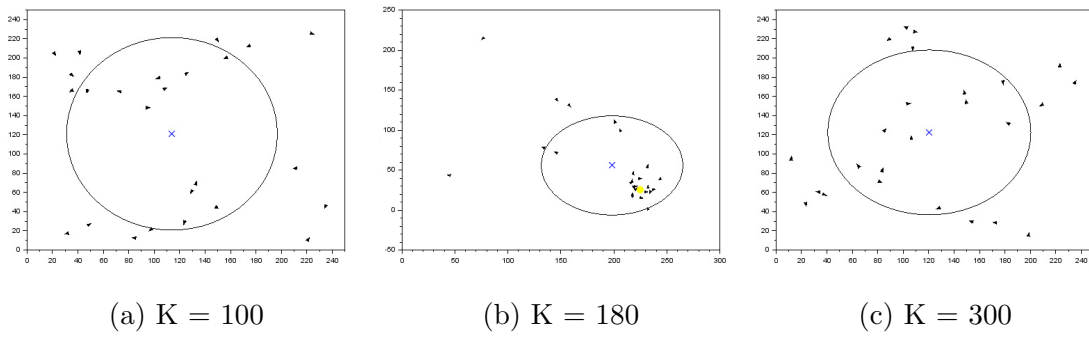
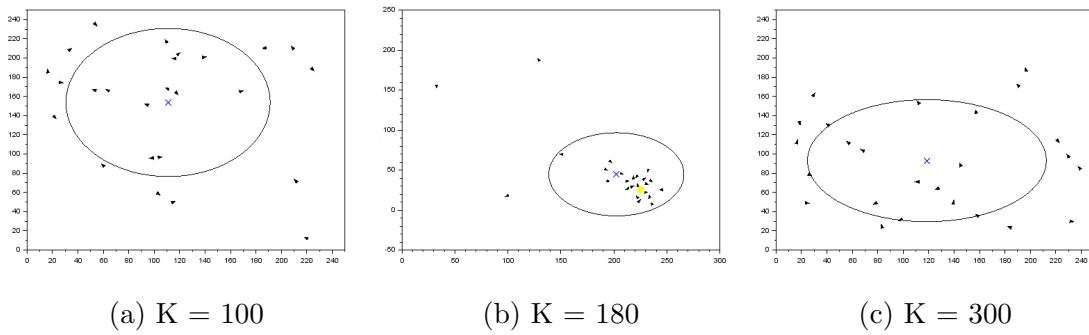


Figura 4.21: Enjambre de robots con $r_r = 5$ y $r_a = 60$, prueba 3

Figura 4.22: Enjambre de robots con $r_r = 10$ y $r_a = 60$, prueba 1Figura 4.23: Enjambre de robots con $r_r = 10$ y $r_a = 60$, prueba 2Figura 4.24: Enjambre de robots con $r_r = 10$ y $r_a = 60$, prueba 3

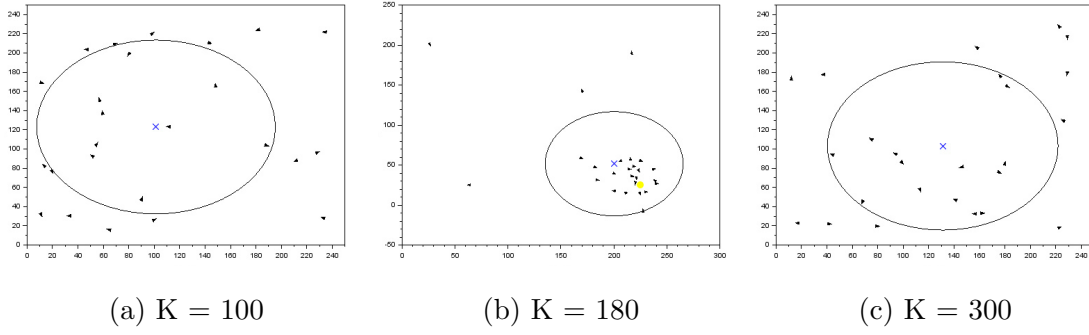


Figura 4.25: Enjambre de robots con $r_r = 15$ y $r_a = 60$, prueba 1

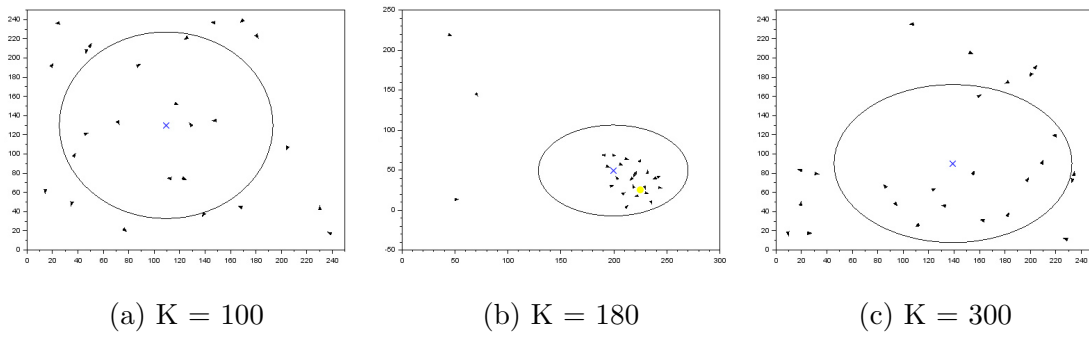


Figura 4.26: Enjambre de robots con $r_r = 15$ y $r_a = 60$, prueba 2

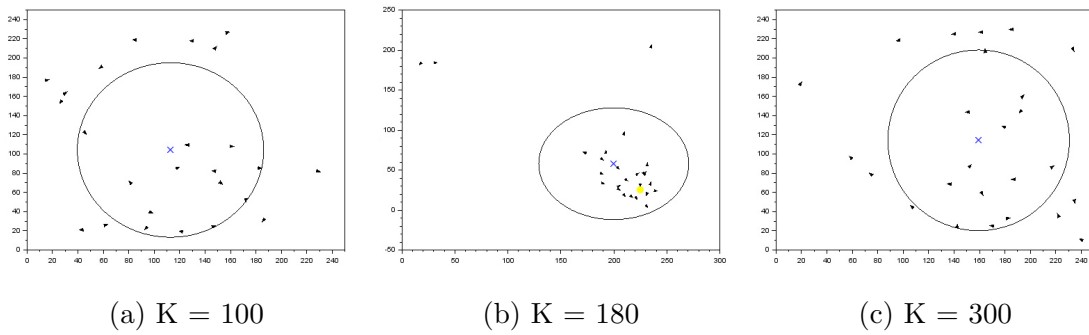


Figura 4.27: Enjambre de robots con $r_r = 15$ y $r_a = 60$, prueba 3

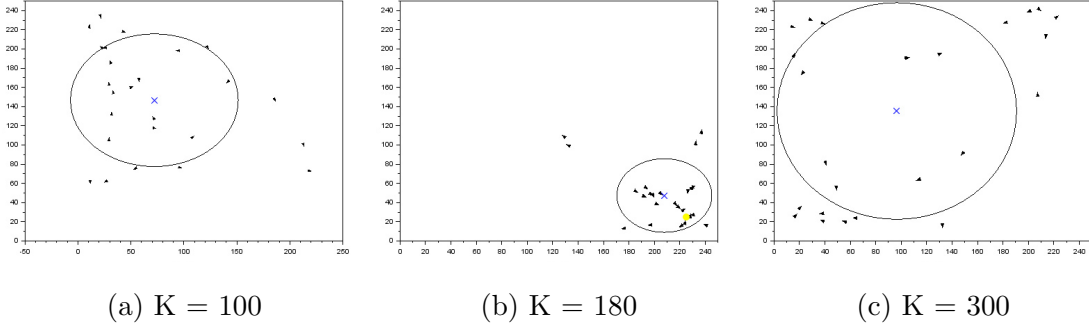


Figura 4.28: Enjambre de robots con $r_r = 5$ y $r_a = 100$, prueba 1

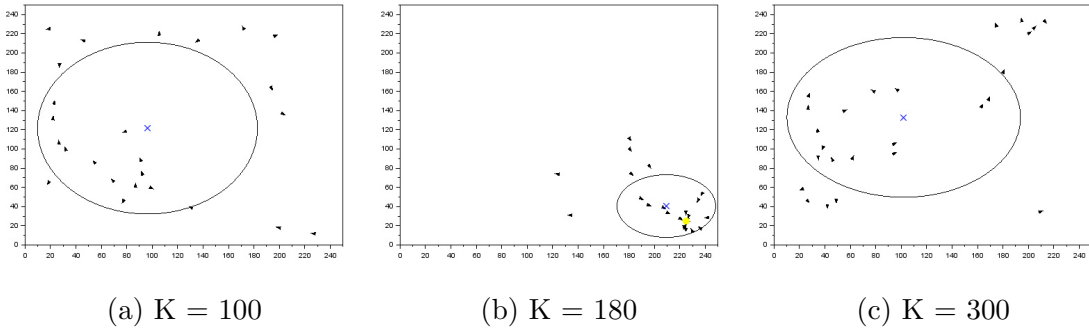


Figura 4.29: Enjambre de robots con $r_r = 5$ y $r_a = 100$, prueba 2

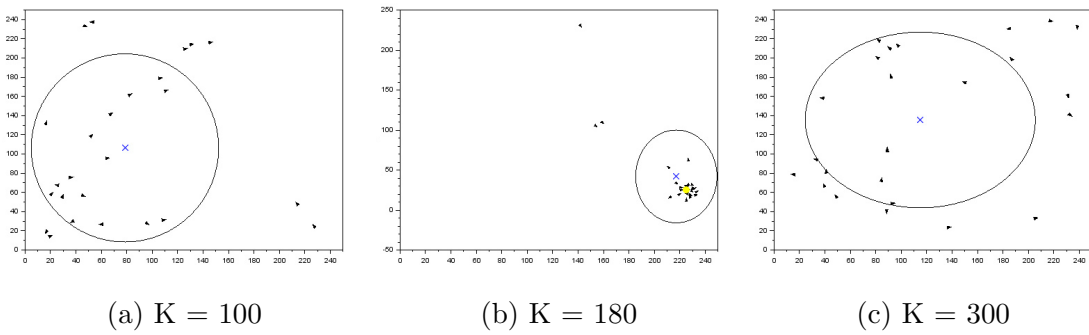


Figura 4.30: Enjambre de robots con $r_r = 5$ y $r_a = 100$, prueba 3

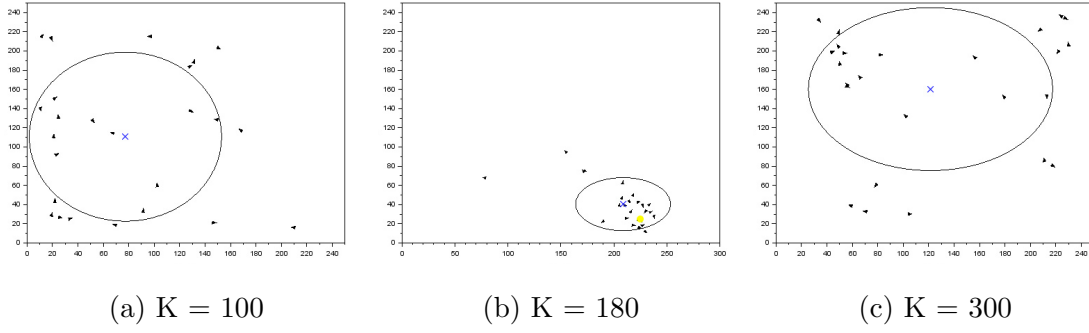


Figura 4.31: Enjambre de robots con $r_r = 10$ y $r_a = 100$, prueba 1

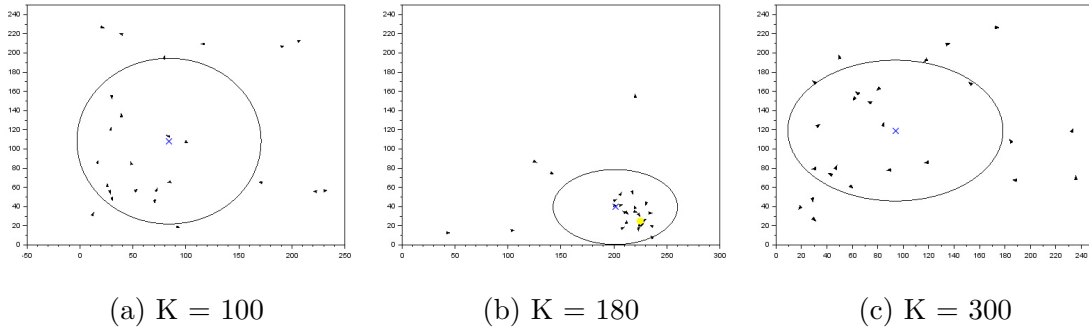


Figura 4.32: Enjambre de robots con $r_r = 10$ y $r_a = 100$, prueba 2

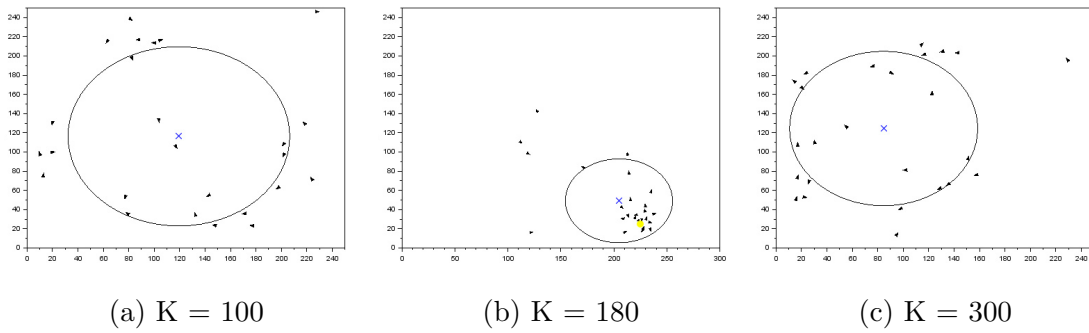


Figura 4.33: Enjambre de robots con $r_r = 10$ y $r_a = 100$, prueba 3

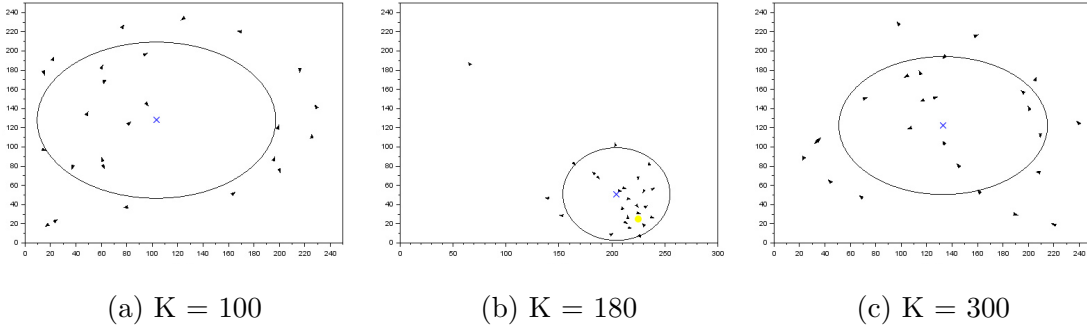


Figura 4.34: Enjambre de robots con $r_r = 15$ y $r_a = 100$, prueba 1

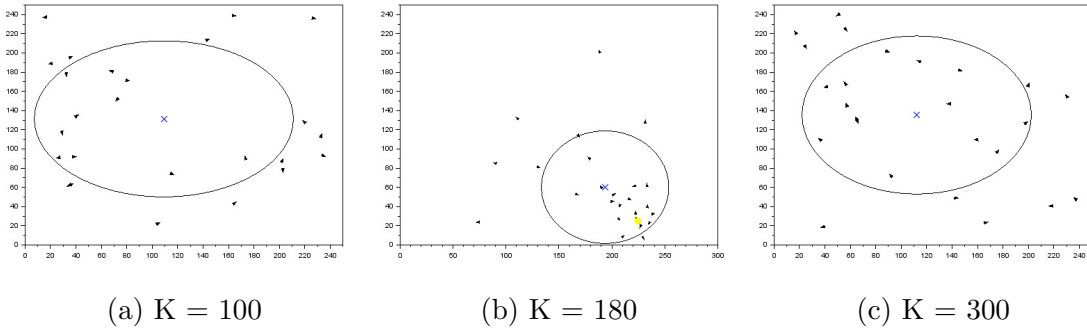


Figura 4.35: Enjambre de robots con $r_r = 15$ y $r_a = 100$, prueba 2

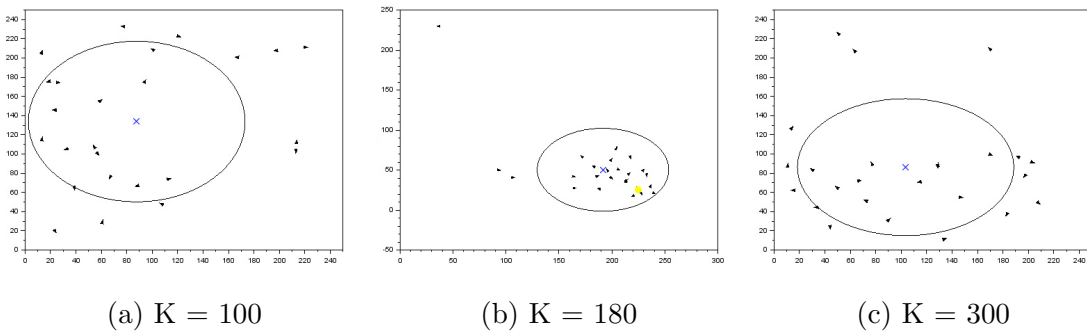


Figura 4.36: Enjambre de robots con $r_r = 15$ y $r_a = 100$, prueba 3

Tabla 4.2: Resultados de primer caso en plataforma de simulación, experimentos del 1 - 3.

No. Exp	r_r	r_a	No. Prueba	K	Cm_x	Cm_y	δ_x	δ_y	Área de elipse
1	5	20	1	100	109.655	120.805	76.477	77.802	18644.394
				180	212.067	44.757	26.642	27.784	2325.504
				300	125.606	121.154	64.981	66.205	13515.323
			2	100	117.327	118.546	72.774	53.021	12121.934
				180	220.053	31.913	21.158	17.764	1180.800
				300	144.331	133.624	54.931	63.869	11021.973
			3	100	137.961	139.527	75.925	61.065	14565.623
				180	220.715	47.964	17.619	34.302	1898.660
				300	113.539	131.230	59.711	70.451	11021.973
			Media	100	121.648	126.293	75.059	63.963	15110.650
				180	217.612	41.545	21.806	26.617	1801.655
				300	127.825	128.669	59.874	66.842	11853.090
2	10	20	1	100	102.612	118.917	69.023	61.541	13344.671
				180	212.574	41.170	30.442	25.863	2473.412
				300	148.033	88.510	59.445	66.083	12341.115
			2	100	133.672	127.211	68.898	70.395	15237.072
				180	215.997	46.823	28.763	30.001	2710.904
				300	128.011	118.447	68.028	62.823	13426.295
			3	100	153.605	111.009	54.593	63.303	10856.907
				180	216.148	38.502	28.158	23.799	2105.251
				300	136.107	78.315	62.413	56.394	11057.602
			Media	100	129.963	119.046	64.172	65.079	13146.217
				180	214.906	42.165	29.121	26.554	2429.856
				300	137.384	95.091	63.295	61.767	12275.004
3	15	20	1	100	133.172	129.817	67.412	65.639	13901.014
				180	209.936	58.296	27.366	41.271	3548.268
				300	157.464	107.520	73.466	69.649	16075.043
			2	100	129.881	134.897	62.098	66.210	12916.680
				180	205.326	42.577	44.396	25.115	3502.871
				300	155.199	85.848	63.509	59.534	11878.266
			3	100	137.265	124.236	63.681	68.663	13736.695
				180	208.850	52.277	33.709	35.302	3738.483
				300	162.984	98.335	55.056	68.393	11829.608
			Media	100	133.439	129.650	64.397	66.837	13518.130
				180	208.037	51.050	35.157	33.896	3596.541
				300	158.549	97.234	64.011	65.859	13260.972

Tabla 4.3: Resultados de primer caso en plataforma de simulación, experimentos del 4 - 6.

No. Exp	r_r	r_a	No. Prueba	K	Cm_x	Cm_y	δ_x	δ_y	Área de elipse
4	5	60	1	100	100.645	140.953	72.191	60.793	13787.466
				180	214.598	36.804	31.104	27.646	2701.477
				300	123.546	108.634	65.299	75.666	15522.174
			2	100	110.235	120.160	76.283	73.513	17617.501
				180	217.650	33.865	32.250	28.481	2885.604
				300	130.757	124.743	65.049	66.657	13621.761
			3	100	105.311	102.108	75.102	54.134	12772.242
				180	222.238	33.161	19.069	23.907	1432.185
				300	134.846	118.331	61.180	67.524	12978.272
			Media	100	105.397	121.074	74.525	62.813	14725.736
				180	218.162	34.610	27.474	26.678	2339.755
				300	129.716	117.236	63.842	69.949	14040.736
5	10	60	1	100	95.642	129.142	67.833	61.890	13189.174
				180	203.823	58.630	40.819	52.106	6681.988
				300	140.382	127.116	73.152	63.274	14541.166
			2	100	113.983	121.019	63.733	77.144	15445.965
				180	198.194	55.684	51.367	47.680	7694.274
				300	120.543	122.536	61.444	66.072	12753.981
			3	100	110.921	153.647	61.781	59.348	11518.819
				180	202.274	44.598	49.048	40.004	6164.181
				300	118.671	93.056	72.291	48.900	11105.659
			Media	100	106.849	134.602	64.449	66.127	13384.653
				180	201.430	52.971	47.078	46.597	6846.814
				300	126.532	114.236	68.962	59.415	12800.269
6	15	60	1	100	101.416	122.940	72.175	69.648	15792.178
				180	200.192	51.559	50.045	50.130	7881.492
				300	131.120	103.147	69.744	67.410	14770.030
			2	100	109.397	129.936	64.605	74.776	15176.551
				180	199.293	49.355	54.344	43.702	7461.084
				300	139.167	90.037	71.947	63.332	14314.647
			3	100	112.822	104.145	56.391	69.946	12391.500
				180	199.804	57.878	54.455	53.765	9197.948
				300	159.316	114.243	54.941	72.432	12501.873
			Media	100	107.878	119.007	64.390	71.456	14453.410
				180	199.763	52.931	52.948	49.199	8180.174
				300	143.201	102.476	65.544	67.725	13862.183

Tabla 4.4: Resultados de primer caso en plataforma de simulación, experimentos del 7 - 9.

No. Exp	r_r	r_a	No. Prueba	K	Cm_x	160.304	δ_x	δ_y	Área de elipse
7	5	100	1	100	72.061	146.668	60.877	53.158	10166.423
				180	207.85	47.379	28.748	29.516	2665.778
				300	96.554	135.422	72.453	86.803	19757.778
			2	100	96.321	121.726	66.511	68.830	14382.091
				180	209.394	40.577	29.845	25.129	2356.145
				300	101.997	132.908	70.668	63.998	14208.087
			3	100	78.603	106.355	56.677	75.414	13427.975
				180	217.239	42.000	24.595	44.632	3448.605
				300	115.053	135.440	69.604	70.414	15397.340
			Media	100	82.328	124.916	61.355	65.800	12658.830
				180	174.318	43.319	27.729	33.093	2823.509
				300	98.551	134.590	70.908	73.738	16454.402
8	10	100	1	100	77.402	110.745	58.183	67.768	12387.022
				180	208.569	40.504	34.440	21.239	2298.018
				300	121.503	160.304	73.883	65.329	15163.433
			2	100	83.985	108.183	66.874	66.417	13953.574
				180	201.213	39.560	45.145	30.047	4261.469
				300	93.900	119.177	65.122	56.447	11548.135
			3	100	119.412	116.327	67.048	71.815	15126.780
				180	204.692	49.173	38.929	33.617	4111.309
				300	84.624	124.467	56.889	61.888	11060.701
			Media	100	93.600	111.752	64.035	68.667	13822.459
				180	204.825	43.079	39.505	28.301	3556.932
				300	100.009	134.649	65.298	61.221	12590.756
9	15	100	1	100	103.325	127.904	72.168	62.614	14195.919
				180	204.186	50.977	38.875	37.123	4533.859
				300	132.975	122.329	63.121	55.285	10963.032
			2	100	109.165	131.382	78.352	62.536	15393.393
				180	193.496	60.217	46.228	45.099	6549.720
				300	112.180	135.447	69.334	63.366	13802.419
			3	100	87.837	133.805	65.513	64.371	13248.451
				180	191.523	50.308	47.759	39.872	5982.385
				300	103.496	86.282	65.460	54.896	11289.289
			Media	100	100.109	131.030	72.011	63.174	14279.254
				180	196.402	53.834	44.287	40.698	5688.655
				300	116.217	114.686	65.972	57.849	12018.247

4.2 RESULTADOS DE SEGUNDO CASO EXPERIMENTAL

4.2.1 EXPERIMENTOS FÍSICOS

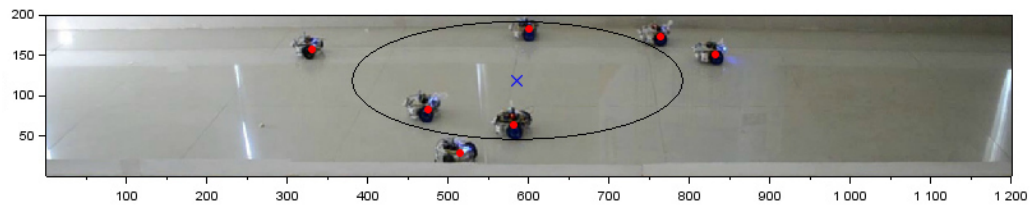
El experimento se realiza con 7 robots en un área de prueba de 1.8 x 6 m. En el plano virtual, el tamaño de la imagen es de 200 x 1200 pixeles. En estos experimentos el radio de orientación se mantiene constante con un valor de 15dm y se repitieron 3 pruebas para cada configuración paramétrica. La duración por experimento es de 1 minuto, y se toma una muestra en el segundo 30. La inclinación en que fueron capturados los experimentos se mantiene, esto para no perder detalles del experimento.

Las Figuras 4.37 - 4.39, corresponde a un radio de atracción es bajo. Esto permite a los robots orientarse en una misma dirección y recorrer el camino hasta la fuente de luz tomando una formación como si fuese un rebaño de ovejas guiadas por un pastor. Al aumentar la repulsión, el enjambre se propaga a lo largo del área de prueba.

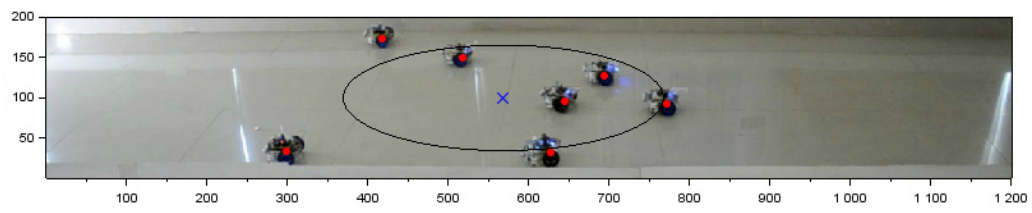
Al aumentar la atracción, ver Figuras 4.40 - 4.42 y 4.43 - 4.45, los robots tienden a alinearse con el vecino de enfrente y la formación se extiende a lo largo del área creando filas de robots. Al aumentar la repulsión, esta alineación se va perdiendo, provocando que los robots abarquen mayor territorio.

A menor repulsión y mayor atracción, los robots llegan a su destino uno tras otro y regresan por el mismo camino. Este comportamiento es similar al de las hormigas al recolectar comida y llevarla a su nido.

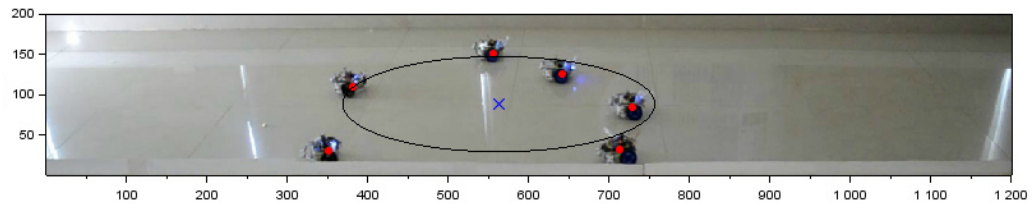
La Tabla 4.5 muestra los datos obtenidos de los experimentos de las Figuras 4.37 a 4.45. Marcando con negrita el valor medio de las tres pruebas realizadas por experimento.



(a) Prueba 1

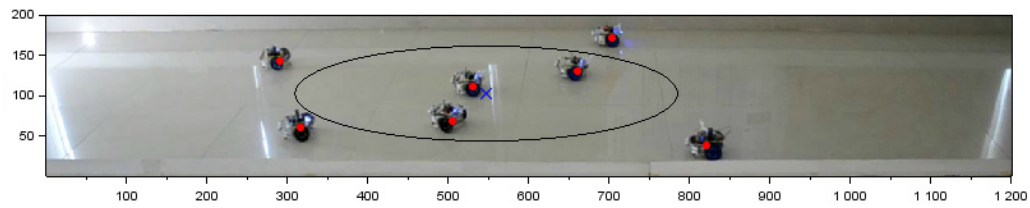


(b) Prueba 2

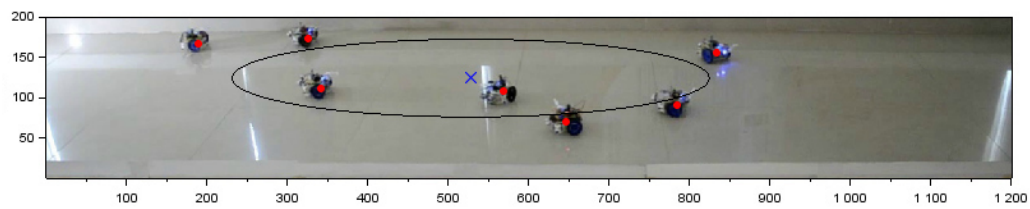


(c) Prueba 3

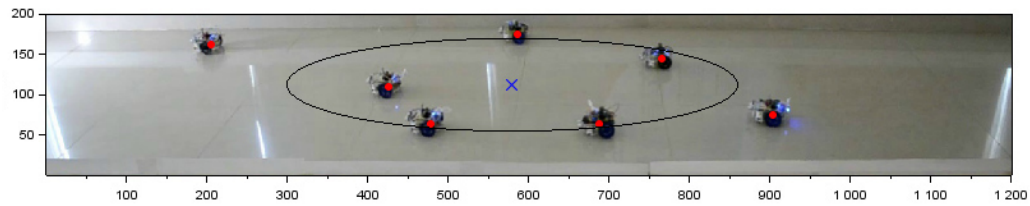
Figura 4.37: Enjambre de robots con $r_r = 5$ y $r_a = 20$



(a) Prueba 1

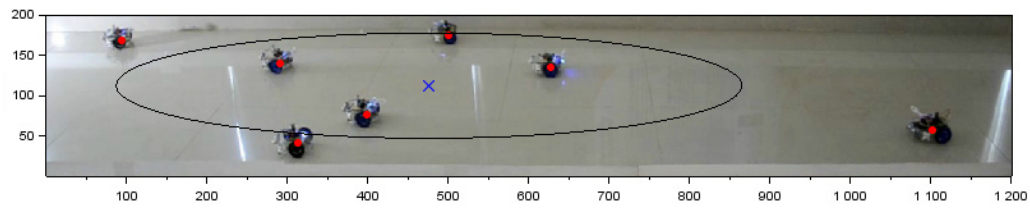


(b) Prueba 2

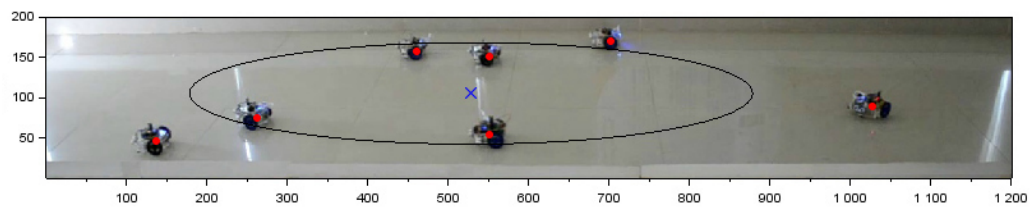


(c) Prueba 3

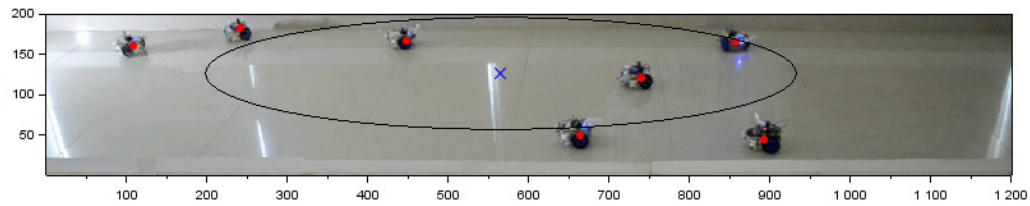
Figura 4.38: Enjambre de robots con $r_r = 10$ y $r_a = 20$



(a) Prueba 1

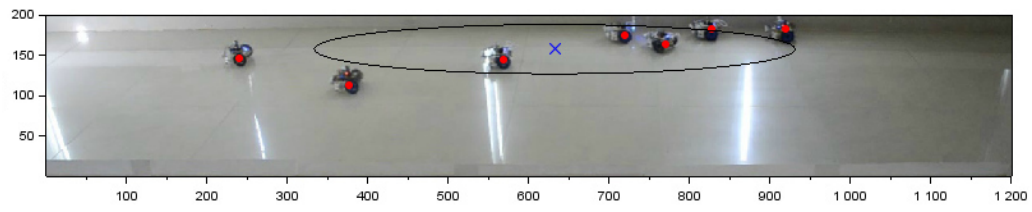


(b) Prueba 2

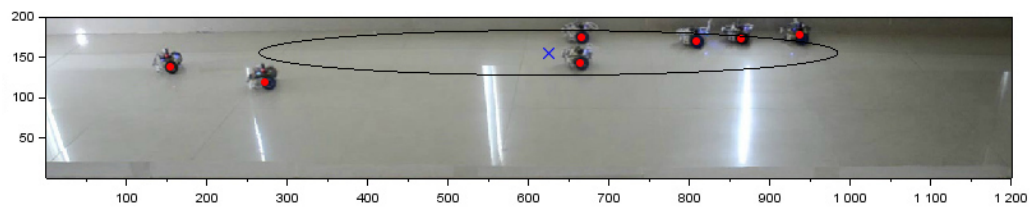


(c) Prueba 3

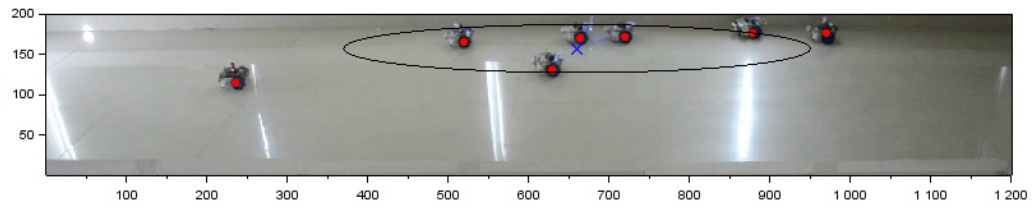
Figura 4.39: Enjambre de robots con $r_r = 15$ y $r_a = 20$



(a) Prueba 1

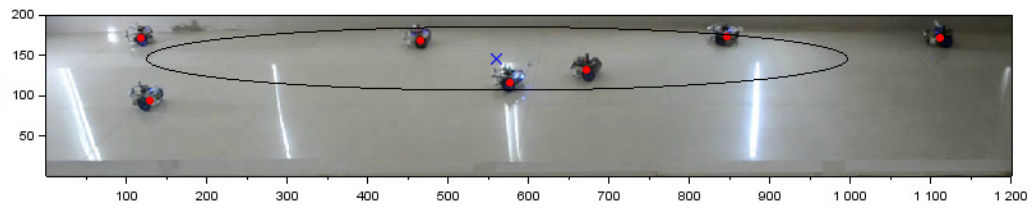


(b) Prueba 2

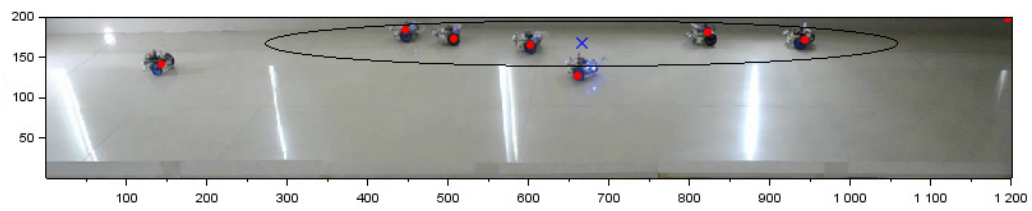


(c) Prueba 3

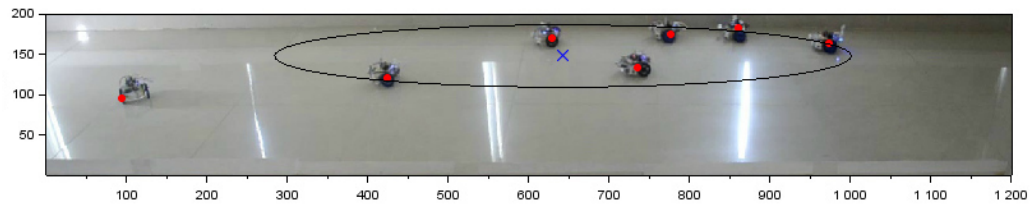
Figura 4.40: Enjambre de robots con $r_r = 5$ y $r_a = 40$



(a) Prueba 1

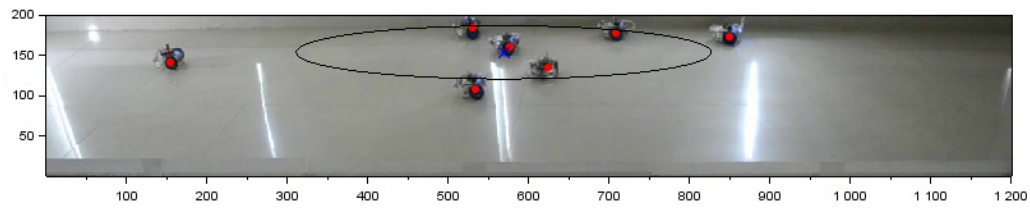


(b) Prueba 2

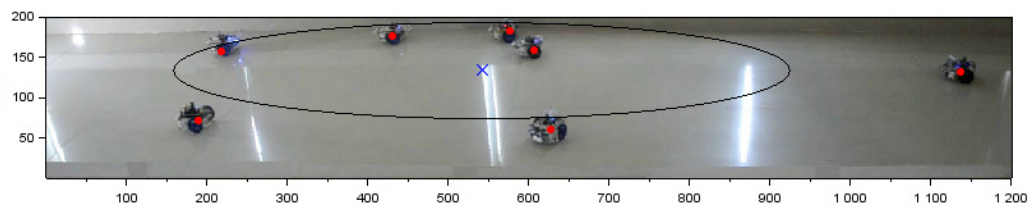


(c) Prueba 3

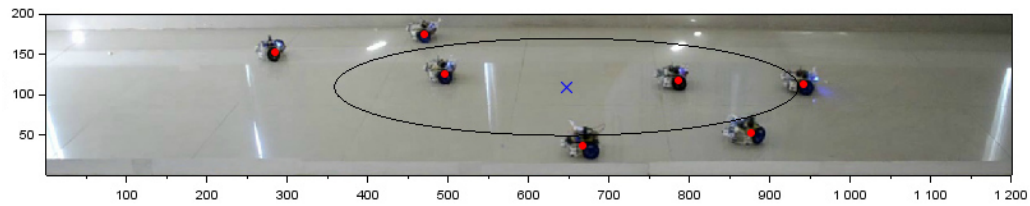
Figura 4.41: Enjambre de robots con $r_r = 10$ y $r_a = 40$



(a) Prueba 1

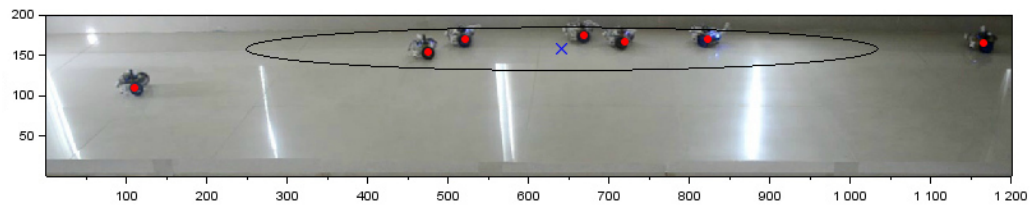


(b) Prueba 2

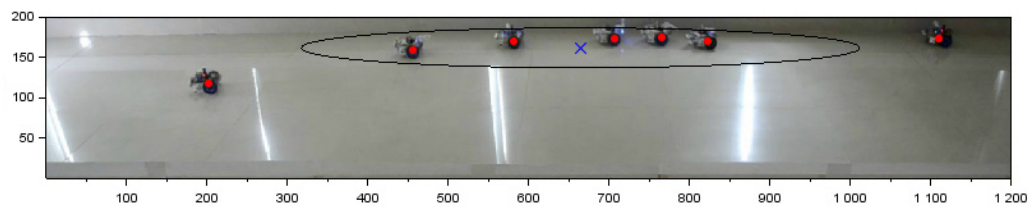


(c) Prueba 3

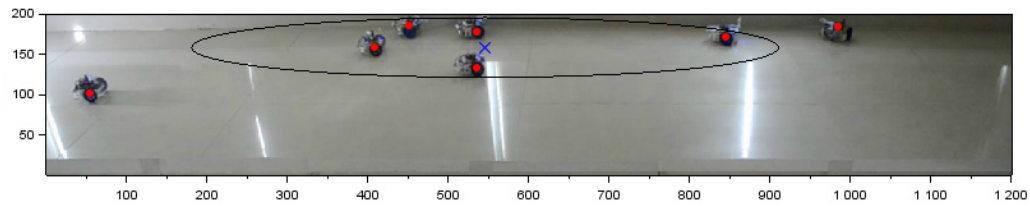
Figura 4.42: Enjambre de robots con $r_r = 15$ y $r_a = 40$



(a) Prueba 1

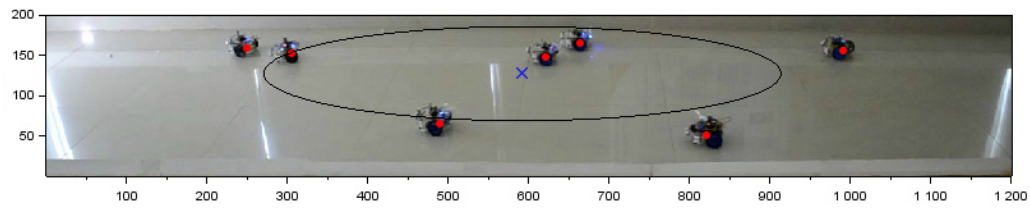


(b) Prueba 2

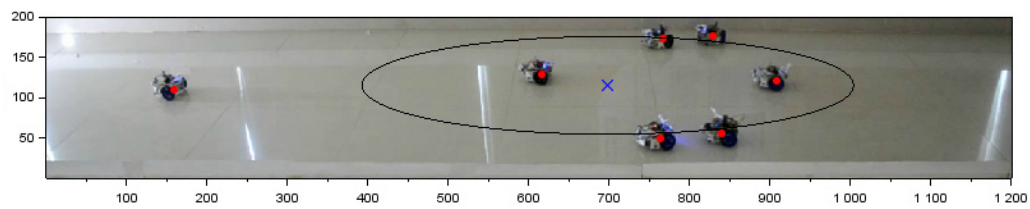


(c) Prueba 3

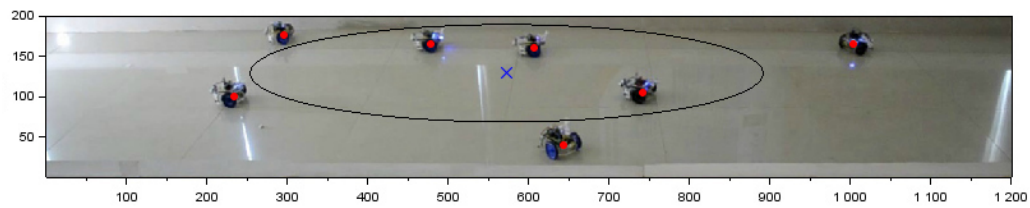
Figura 4.43: Enjambre de robots con $r_r = 5$ y $r_a = 60$



(a) Prueba 1

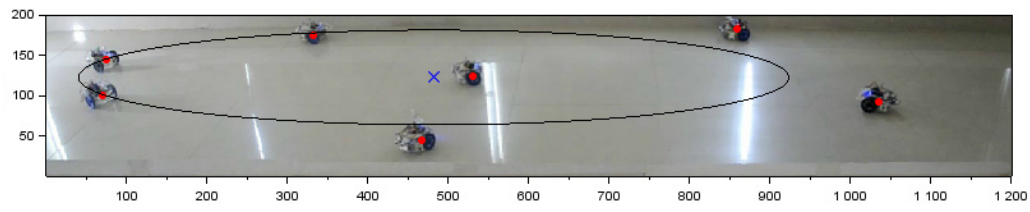


(b) Prueba 2

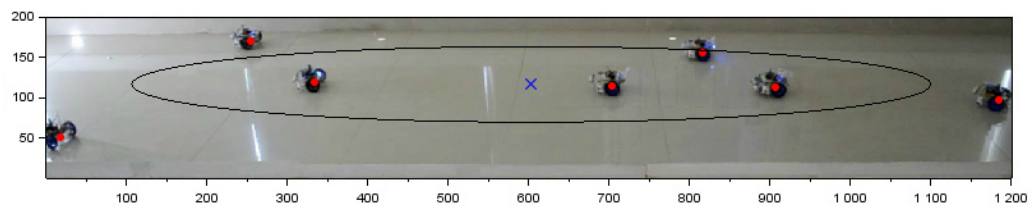


(c) Prueba 3

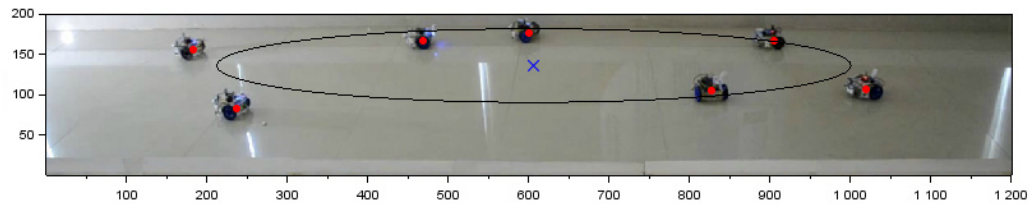
Figura 4.44: Enjambre de robots con $r_r = 10$ y $r_a = 60$



(a) Prueba 1



(b) Prueba 2



(c) Prueba 3

Figura 4.45: Enjambre de robots con $r_r = 15$ y $r_a = 60$

Tabla 4.5: Resultados de segundo caso experimental con robots físicos.

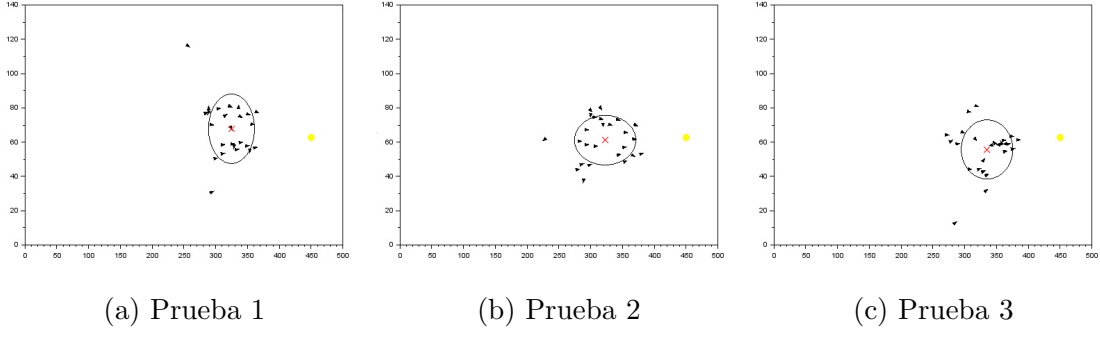
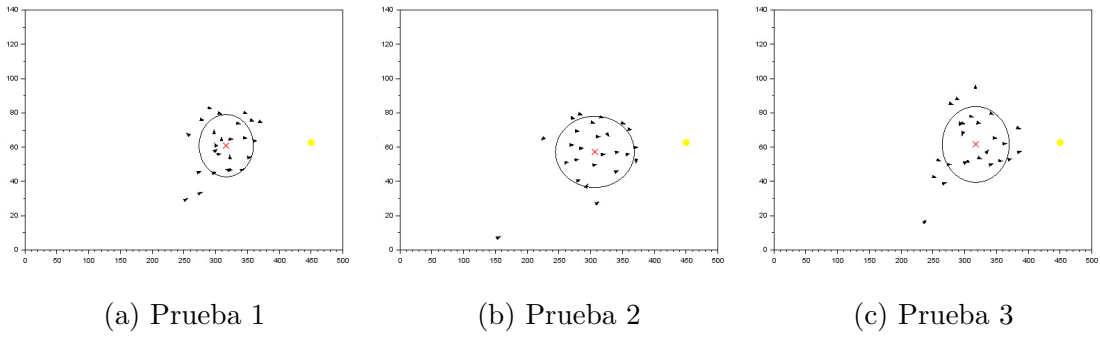
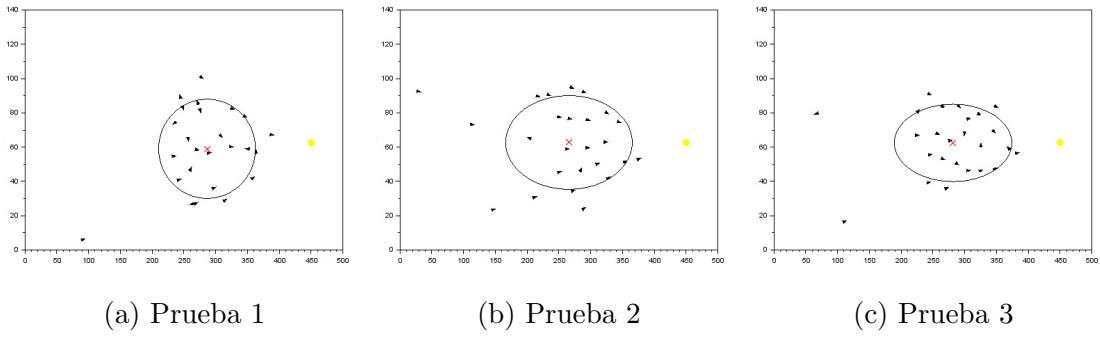
No. Exp	r_r	r_a	No. Prueba	Cm_x	Cm_y	δ_x	δ_y	Área de elipse
1	5	20	1	586.214	119.071	157.971	56.058	27820.282
			2	568.571	100.000	153.159	50.344	24223.515
			3	563.083	88.500	149.273	45.453	21315.469
			Media	572.623	102.524	153.467	50.618	24453.089
2	10	20	1	547.500	102.929	182.954	45.276	26022.903
			2	528.143	124.500	228.295	37.107	26613.521
			3	579.714	112.571	215.629	44.187	29932.887
			Media	551.786	113.333	208.959	42.190	27523.104
3	15	20	1	476.357	112.714	299.304	50.247	47246.975
			2	528.571	105.571	269.361	48.479	41024.268
			3	565.500	126.643	282.531	53.388	47386.878
			Media	523.476	114.976	283.732	50.705	45219.374
4	5	40	1	632.429	157.929	229.874	23.524	16988.509
			2	624.643	155.929	277.031	21.343	18575.319
			3	660.071	157.357	223.250	22.864	16035.957
			Media	639.048	157.071	243.385	22.577	17199.928
5	10	40	1	569.143	154.000	198.603	25.383	15837.067
			2	666.063	167.438	302.554	21.618	20548.008
			3	642.714	148.143	275.719	29.727	25749.390
			Media	625.973	156.527	258.959	25.576	20711.488
6	15	40	1	560.786	146.214	335.499	30.237	31869.830
			2	541.857	133.857	294.756	45.801	42412.223
			3	647.000	109.786	221.940	46.328	32301.907
			Media	583.214	129.952	284.065	40.789	35527.987
7	5	60	1	641.357	158.214	302.051	20.506	19458.414
			2	664.500	162.000	267.170	19.021	15964.798
			3	545.929	158.714	280.763	28.492	25130.920
			Media	617.262	159.643	283.328	22.673	20184.711
8	10	60	1	592.357	127.571	247.430	44.874	34881.654
			2	698.500	115.500	235.304	46.628	34469.119
			3	572.714	129.357	245.443	46.409	35785.446
			Media	621.190	124.143	242.726	45.971	35045.406
9	15	60	1	482.000	122.929	339.954	45.181	48253.050
			2	603.357	116.786	382.247	36.081	43328.473
			3	606.571	136.357	303.438	35.142	33500.364
			Media	563.976	125.357	341.879	38.801	41693.962

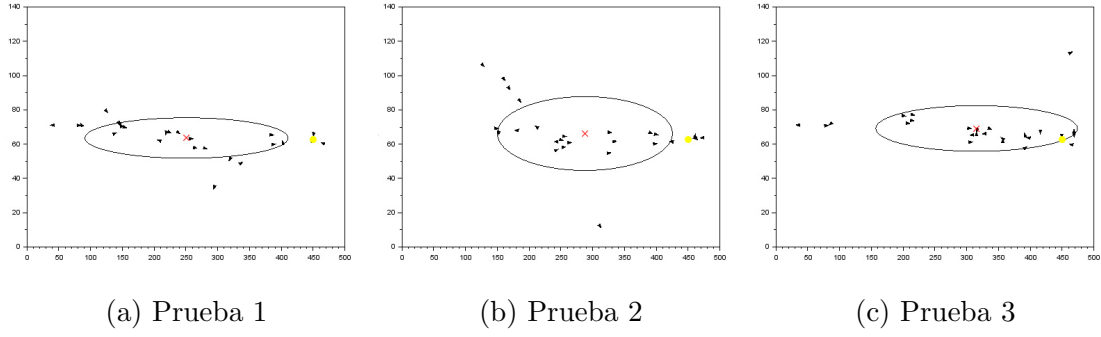
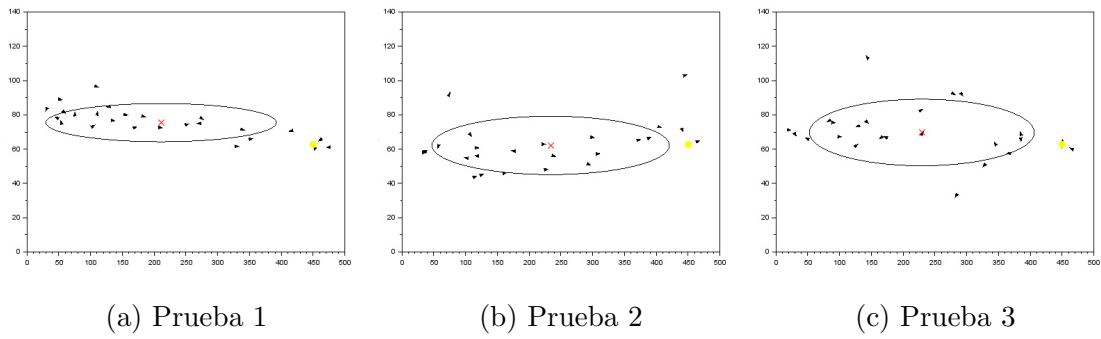
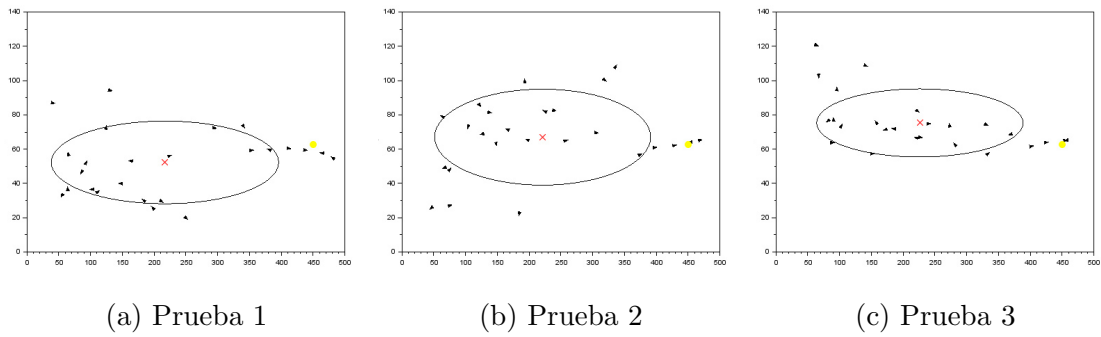
4.2.2 EXPERIMENTOS POR SIMULACIÓN

Las simulaciones se realizaron con 25 individuos en un área de prueba de 14 x 50 m. Las gráficas presentadas se muestran en unidad de decímetros. En estos experimentos el radio de orientación se mantiene constante con un valor de $15dm$. Se simula una fuente de luz colocándola en las coordenadas $[450,70]$ del área de prueba. Las simulaciones se repitieron 3 veces para cada configuración paramétrica. Por cada simulación se hace un total de 300 iteraciones, y se toma una muestra en la iteración 150.

Las Figuras 4.46 - 4.48 corresponden a una atracción de $20dm$, donde se observa una formación similar obtenida en los experimentos físicos con atracción baja, orientando a los individuos en dirección de la zona de influencia siendo guiados por esta como un rebaño de ovejas. Al aumentar la atracción a $60dm$ y $100dm$, ver Figuras 4.49 - 4.51 y 4.52 - 4.54, el enjambre tiende a visitar la zona influenciada regresando al origen y volviendo a esta. Al aumentar la repulsión, el enjambre tarda más tiempo en llegar al punto deseado.

La Tabla 4.6 muestra los datos obtenidos de los experimentos de las Figuras 4.46 - 4.54, marcando con negrita el valor medio de las tres pruebas realizadas por experimento.

Figura 4.46: Enjambre de robots con $r_r = 5$ y $r_a = 20$ Figura 4.47: Enjambre de robots con $r_r = 10$ y $r_a = 20$ Figura 4.48: Enjambre de robots con $r_r = 15$ y $r_a = 20$

Figura 4.49: Enjambre de robots con $r_r = 5$ y $r_a = 60$ Figura 4.50: Enjambre de robots con $r_r = 10$ y $r_a = 60$ Figura 4.51: Enjambre de robots con $r_r = 15$ y $r_a = 60$

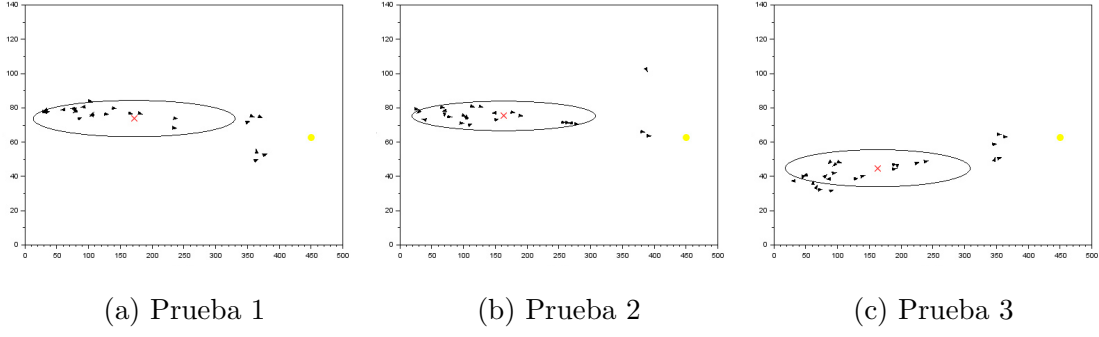
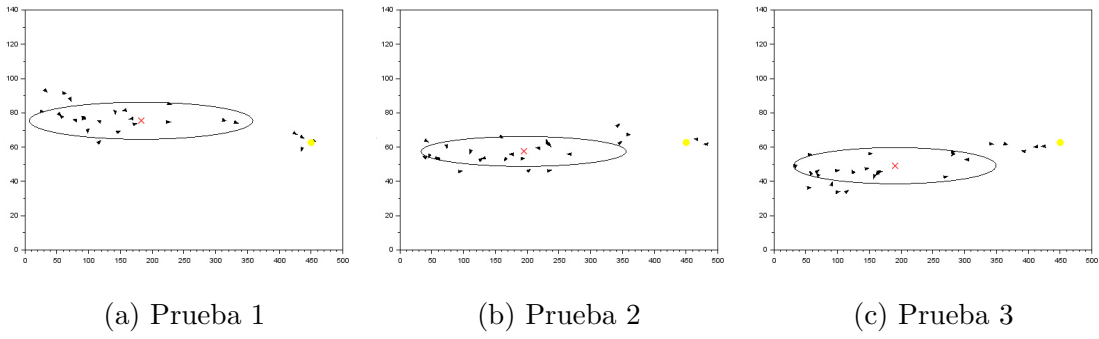
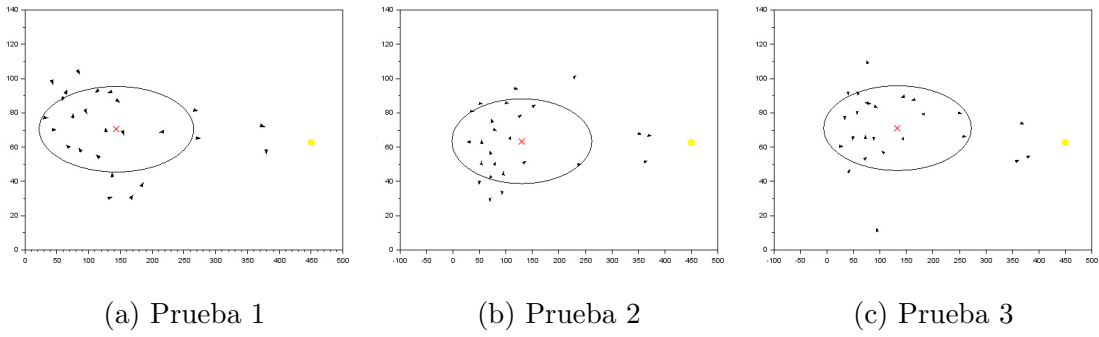
Figura 4.52: Enjambre de robots con $r_r = 5$ y $r_a = 100$ Figura 4.53: Enjambre de robots con $r_r = 10$ y $r_a = 100$ Figura 4.54: Enjambre de robots con $r_r = 15$ y $r_a = 100$

Tabla 4.6: Resultados de segundo caso experimental en plataforma de simulación.

No. Exp	r_r	r_a	No. Prueba	Cm_x	Cm_y	δ_x	δ_y	Área de elipse
1	5	20	1	324.679	67.823	27.793	15.587	1360.941
			2	334.947	55.743	31.175	13.271	1299.741
			3	322.651	61.165	37.084	11.120	1295.464
			Media	327.426	61.577	32.018	13.326	1318.715
2	10	20	1	316.463	60.864	32.781	13.956	1437.286
			2	306.571	57.262	47.782	15.997	2401.393
			3	317.242	61.575	40.566	17.018	2168.786
			Media	313.425	59.901	40.376	15.657	2002.488
3	15	20	1	286.346	59.039	58.580	22.254	4095.578
			2	265.674	62.713	76.531	21.019	5053.624
			3	281.559	62.480	71.051	17.345	3871.536
			Media	277.860	61.411	68.720	20.206	4340.246
4	5	60	1	250.794	63.573	122.924	9.089	3509.782
			2	287.602	66.197	105.896	16.593	5520.057
			3	315.455	69.149	121.956	10.256	5520.057
			Media	284.617	66.306	116.926	11.979	4319.763
5	10	60	1	210.903	75.463	139.596	8.587	3765.850
			2	233.690	62.188	143.634	13.105	5913.560
			3	229.352	69.711	136.065	14.939	6385.716
			Media	224.648	69.121	139.765	12.210	5355.042
6	15	60	1	216.950	52.266	137.535	18.562	8020.437
			2	221.103	67.010	130.617	21.530	8834.575
			3	226.193	75.292	124.828	15.293	5997.157
			Media	221.415	64.856	130.994	18.462	7617.390
7	5	100	1	171.698	73.728	122.058	8.162	3129.704
			2	163.062	75.279	111.054	6.627	2311.924
			3	163.167	44.777	111.894	8.320	2924.592
			Media	165.976	64.594	115.002	7.703	2788.740
8	10	100	1	182.498	75.382	135.219	8.313	3531.463
			2	194.510	57.483	124.142	6.708	2615.996
			3	190.052	49.176	122.654	8.125	3130.696
			Media	189.020	60.680	127.338	7.715	3092.718
9	15	100	1	143.730	70.450	93.434	19.197	5635.070
			2	129.886	63.461	101.507	19.025	6066.780
			3	133.093	71.103	107.147	19.012	6399.602
			Media	135.570	68.338	100.696	19.078	6033.818

CAPÍTULO 5

CONCLUSIONES

Este trabajo tiene como objetivo el análisis del comportamiento emergente en enjambres de robots, para esto se toma como base el modelo de comportamiento de seres sociales propuesto por Ian Couzin, adaptándolo a un enjambre de robots simples que utilizan solo información local percibida por sus sensores.

Se desarrolló una plataforma de simulación para validar el comportamiento del enjambre robótico con una cantidad mayor de individuos, considerando las limitaciones presentes en los prototipos de robots implementados. Las simulaciones se realizaron con distintas configuraciones paramétricas, dando como resultado una variación en el comportamiento emergente del enjambre.

En base a las pruebas realizadas, cuando el enjambre no es afectado por factores externos, los individuos exploran el área de prueba bajo las reglas de comportamiento implementadas, manteniendo una misma dirección hasta llegar a un límite o al encontrarse con sus vecinos y formar grupos. El cambio de valores de repulsión y atracción afecta en la formación y comportamiento del enjambre. El utilizar valores altos de repulsión eleva la dispersión del enjambre, mientras que valores bajos de repulsión hace que los miembros se mantengan más cercanos. La atracción permite a los individuos encontrar con mayor rapidez a sus vecinos y formar grupos.

Se puede influir a este tipo estructuras colectivas utilizando métodos descentralizados para atraer o alejar al enjambre de un punto interés, teniendo la capacidad de cambiar de una estructura a otra y volver sin modificar los parámetros de los agentes. Los comportamientos que emergen al interactuar con algún tipo de influencia pueden tener aplicación en diversas tareas, además de modificar la formación con solo realizar cambios en ciertos parámetros. Se observa que la orientación del enjambre es gobernada por el nivel de influencia en el entorno, donde cada individuo busca la zona con mayor intensidad de luz.

Las pruebas realizadas, al implementar las reglas de comportamiento en un sistema embebido, demuestran que es posible gobernar en el enjambre sin requerir información global o un control centralizado.

Debido a las limitaciones sensoriales de los robots, el comportamiento que surge en los experimentos físicos no corresponde en su totalidad con el observado en las simulaciones. Sin embargo, los resultados obtenidos son similares en la mayor parte de tiempo.

Como trabajo futuro en este estudio, queda abierta la exploración del comportamiento con una influencia variable o negativa. Otro tema a desarrollar es la colaboración en el enjambre para realizar una tarea particular. Además de la adaptación de este modelo a reglas de comportamiento más cercanas a la realidad, con el fin de que los robots puedan diferenciar entre sus vecinos y los obstáculos presentados en el entorno o que estos puedan comunicar información sobre algún objetivo.

APÉNDICE A

REPORTE TÉCNICO DE LOS ROBOTS.

A.1 DISEÑO

La estructura del robot está formada por dos placas de acrílico con un diámetro de 12.5 cm. Entre las dos placas se encuentran los motores y la batería. Las ruedas son de goma para evitar el deslizamiento y tienen un diámetro de 6 cm. La placa de circuito impreso se ubica sobre la estructura. Una vista de la estructura mecánica puede verse en la Figura A.1.

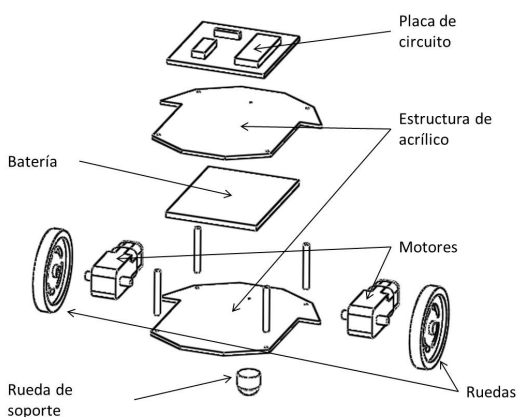


Figura A.1: Vista explosionada de la estructura del robot

A.2 HARDWARE

A.2.1 MICROCONTROLADOR

El procesamiento del robot es realizado por una placa Arduino Micro, la cual incorpora un microcontrolador de 8 bits, comunicación USB y todos los elementos necesarios para su funcionamiento. La Tabla A.1 muestra un resumen de sus las características.

Tabla A.1: Características de Arduino Micro

Microcontrolador	ATmega32u4
Voltaje de operación	5 V
Pines digitales I/O	20
Canales PWM	7
Canales de entrada analógica	12
Corriente DC por pin I/O	40 mA
Memoria Flash	32 KB (ATmega32u4)
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Velocidad de reloj	16 MHz

A.2.2 SENSORES Y ACTUADORES

Al frente del robot se ubica un sensor ultrasónico HC-SR04, cuyas características se muestran en la Tabla A.2.

A la derecha e izquierda del robot se utilizan dos sensores infrarrojos TCRT5000, con un rango aproximado de detección de 40 cm.

Tabla A.2: Características del sensor ultrasónico

Ángulo efectivo	<20 grados
Rango de distancia	2 cm - 400 cm
Resolución	0.3 cm

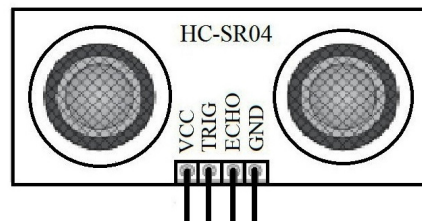


Figura A.2: Asignación de pines del sensor ultrasónico HC-SR04

Tabla A.3: Características de los sensores infrarojos

Voltaje de polarización del emisor	1.5 V
Corriente del emisor	60 mA
Longitud de onda del emisor	950 nm
Distancia para operación pico	2.5 mm

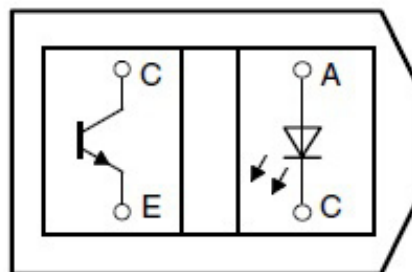


Figura A.3: Asignación de pines del sensor infrarrojo TCRT5000.

Cada robot utiliza dos motores de corriente directa con reducción de velocidad y un circuito integrado SN754410 para controlarlos. Las características de los motores son las siguientes:

Tabla A.4: Características de los motores

Voltaje de operación	3 V - 12 V
Relación de reducción	120:1
Velocidad sin carga	170 rpm (3 V)
Consumo de corriente	70 mA (250 mA max.)
Par	800 gf/cm

A.2.3 ALIMENTACIÓN

Cada robot cuenta con una batería recargable Li-ion de 4.1 V y un módulo (Figura A.4) para elevar la tensión a 5V, con salida USB para alimentar la placa Arduino. Este módulo también se utiliza para cargar la batería, alimentándolo con 5V mediante el puerto micro USB.

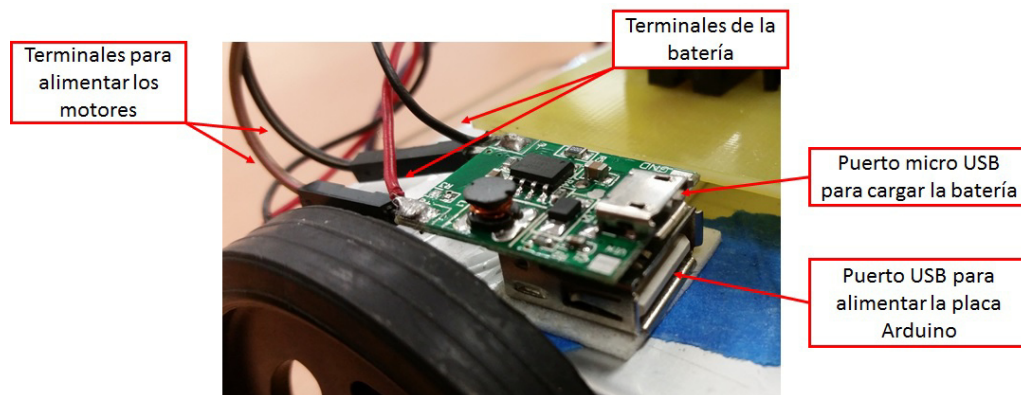


Figura A.4: Módulo convertidor DC-DC y cargador de batería.

La Figura A.5 muestra el diagrama esquemático del circuito impreso que controla cada robot.

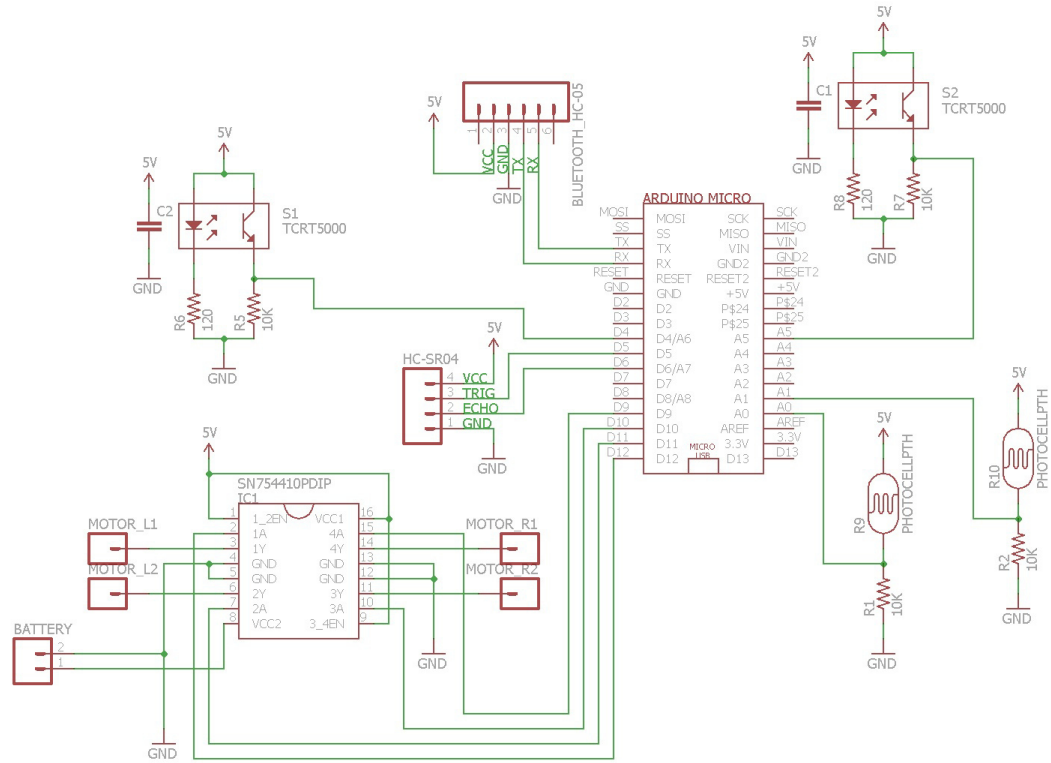


Figura A.5: Diagrama esquemático

A.3 SOFTWARE

La programación del microcontrolador se realiza en el entorno de desarrollo integrado de *Arduino* (Arduino IDE 1.6.8).

Para la comunicación bluetooth con la computadora, se utiliza la aplicación *Bluetooth Serial Terminal* para *Windows 10*, mientras que para la comunicación con un celular se utiliza la aplicación *Bluetooth Terminal* para *Android*.

A.4 FUNCIONAMIENTO

Para encender los robots es necesario conectar un cable USB - micro USB del módulo de la batería a la placa Arduino y presionar el botón de inicialización como se muestra en la figura A.6.

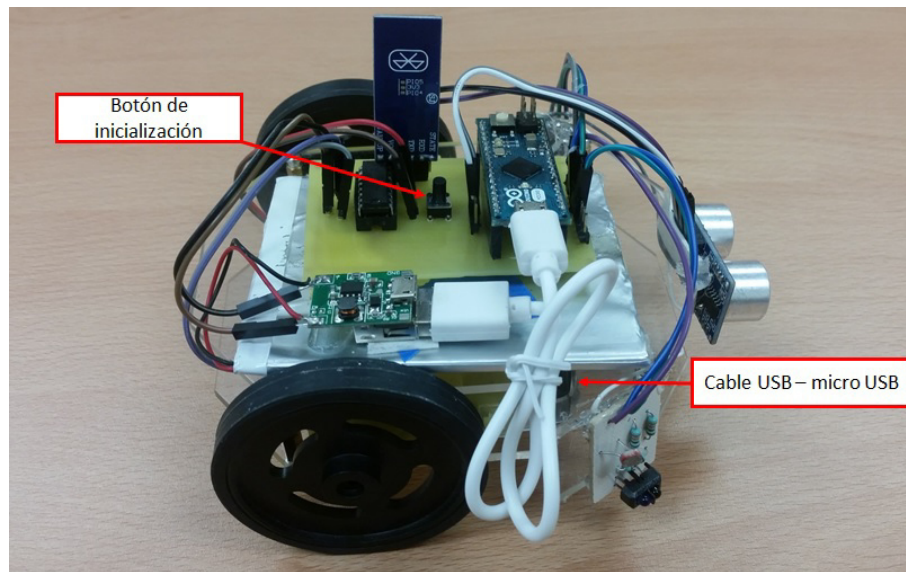


Figura A.6: Alimentación del Arduino y los motores

Una vez encendido, el robot entra en modo automático hasta que reciba un comando por el puerto serie. Los comandos utilizados para el control del robot se muestran en el Apéndice B.

APÉNDICE B

CÓDIGO EN ARDUINO - REGLAS DE COMPORTAMIENTO

```
1  //*****
2  //Estudio de comportamintos emergentes en enjambres de robots basados
3  //factores de influencia
4  //MCIE Mecatronica
5  //UANL – FIME
6  //Erick de Jesus Ordaz Rivas – Mario Aguilera Ruiz
7  //*****
8
9  //libraries
10 #include <NewPing.h>
11 #include <SoftwareSerial.h>
12
13 //definition of pins I/O
14 #define TRIGGER_PIN 5 //digital output for ultrasonic sensor.
15 #define ECHO_PIN 6 //digital input for ultrasonic sensor.
16 int button = 4; //digital input for start button.
17 int motorRD = 12; //digital output for right engine.
18 int motorLD = 10; //digital output for left engine.
19 int motorRA = 11; //analog output for right engine.
20 int motorLA = 9; //analog output for left engine.
21 int ldrD = A2; //analog input for left LDR.
```

```
22  int ldrI = A4;           //analog input for right LDR.
23  int irD = A3;           //analog input for right infrared sensor.
24  int irI = A5;           //analog input for left infrared sensor.
25
26  //initialization function for ultrasonic sensor.
27  NewPing sonar(TRIGGER_PIN, ECHO_PIN, 400);
28
29  //definition of variables and constants.
30  int statebutton = 0;    //activation button state.
31  int previousstate = 0; //previous state of button.
32  int swarmon = 0;        //initialization flag.
33  int velRr, velRl;       //engine speed (repulsion).
34  int velAr, velAl;       //engine speed (attraction).
35  int velr, vell, velP;   //engine speed.
36  int val;               //value read by serial port.
37  int distance;          //distance calculated by ultrasonic sensor.
38  int blue = 1;          //bluetooth flag.
39  int ldrI_val, ldrD_val; //variables for reading of LDR.
40  int irI_val, irD_val;  //variables for reading of infrared sensor.
41  int rr=5;              //distance of repulsion.
42  int ro=15;             //distance of orientation.
43  int ra=40;             //distance of attraction.
44  int Qf[]={0, 1};       //vector in direction of ultrasonic sensor.
45  int Ql[]={-1, 0};      //vector in direction of left infrared sensor.
46  int Qr[]={1, 0};       //vector in direction of right infrared sensor.
47  int q1a                //detected items by sesnors in zone of attraction.
48  int q1r,q2r,q3r;       //detected items by sensors in zone of repulsion.
49  int detR;              //detected items in zone of repulsion.
50  int detO;              //detected items in zone of orientation.
51  int detA;              //detected items in zone of attraction.
52  int DQ1;               //distance measured by ultrasonic sensor.
53  int DQ2;               //distance measured by left infrared sensor.
54  int DQ3;               //distance measured by right infrared sensor.
55  int DLl;               //amount of light received by left LDR.
56  int DLr;               //amount of light received by right LDR.
57  int ddx,ddy;           //components of desired direction.
```

```
58  int dd,dd2;           //desired direction.
59
60  void setup() {
61    //start serial communication.
62    Serial.begin(9600);
63    Serial1.begin(9600);
64    //pin configuration of engines.
65    pinMode(motorLD, OUTPUT);
66    pinMode(motorRD, OUTPUT);
67    pinMode(motorLA, OUTPUT);
68    pinMode(motorRA, OUTPUT);
69    digitalWrite(button);
70  }
71
72  void loop() {
73    if (Serial.available()){
74      blue = '0'; //check for commands.
75    }
76    if (Serial1.available()){
77      blue = '0'; //check for commands.
78    }
79    while (blue == '0') {
80      if (Serial1.available()) {
81        val = Serial1.read(); //read command.
82      }
83      if (Serial.available()) {
84        val = Serial.read(); //read command.
85      }
86      if (val == 'F') { //manual mode-advance.
87        Forward();
88      }
89      if (val == 'B') { //manual mode-reverse.
90        Backward();
91      }
92      if (val == 'L') { //manual mode-turn left.
93        TurnLeft(5);
```



```
94     Stop();
95 }
96 if (val == 'R') { //manual mode-turn right.
97     TurnRight(5);
98     Stop();
99 }
100 if (val == 'S') { //manual mode-stop.
101     Stop();
102 }
103 if (val == 'X') { //change manual mode to automatic.
104     blue = '1';
105 }
106
107 //change of parameter value.
108 if (val == '1') {
109     Stop();
110     rr=5;
111     ro=15;
112     ra=20;
113     blue = '1';
114 }
115 if (val == '2') {
116     Stop();
117     rr=10;
118     ro=15;
119     ra=20;
120     blue = '1';
121 }
122 if (val == '3') {
123     Stop();
124     rr=15;
125     ro=15;
126     ra=20;
127     blue = '1';
128 }
129 if (val == '4') {
```

```
130      Stop();
131      rr=5;
132      ro=15;
133      ra=40;
134      blue = '1';
135  }
136  if ( val == '5' ) {
137      Stop();
138      rr=10;
139      ro=15;
140      ra=40;
141      blue = '1';
142  }
143  if ( val == '6' ) {
144      Stop();
145      rr=15;
146      ro=15;
147      ra=40;
148      blue = '1';
149  }
150  if ( val == '7' ) {
151      Stop();
152      rr=5;
153      ro=15;
154      ra=60;
155      blue = '1';
156  }
157  if ( val == '8' ) {
158      Stop();
159      rr=10;
160      ro=15;
161      ra=60;
162      blue = '1';
163  }
164  if ( val == '9' ) {
165      Stop();
```

```
166         rr=15;
167         ro=15;
168         ra=60;
169         blue = '1';
170     }
171 } //end while blue == 0
172
173 statebutton = digitalRead(button);
174 if ((statebutton == HIGH) && (previousstate == LOW)){
175     swarmon = 1 - swarmon;
176 }
177 previousstate = statebutton;
178
179 //sensor readings.
180 ldrI_val = analogRead(ldrI);
181 ldrD_val = analogRead(ldrD);
182
183 int irI_val1 = analogRead(irI);
184 int irD_val1 = analogRead(irD);
185 int irI_val2 = analogRead(irI);
186 int irD_val2 = analogRead(irD);
187 int irI_val3 = analogRead(irI);
188 int irD_val3 = analogRead(irD);
189 int irI_val4 = analogRead(irI);
190 int irD_val4 = analogRead(irD);
191 int irI_val5 = analogRead(irI);
192 int irD_val5 = analogRead(irD);
193 int irI_val6 = analogRead(irI);
194 int irD_val6 = analogRead(irD);
195
196 irI_val=(irI_val1 + irI_val2 + irI_val3 + irI_val4 + irI_val5 + irI_val6)/6;
197 irD_val=(irD_val1 + irD_val2 + irD_val3 + irD_val4 + irD_val5 + irD_val6)/6;
198
199 DQ2 = 126.43*pow(irI_val, -0.692);
200 DQ3 = 126.43*pow(irD_val, -0.692);
201 DQ2 = constrain(DQ1, 1, 20);
```

```

202   DQ3 = constrain(DQr, 1, 20);
203
204   DQ1 = Ultrasonico(); //reading of ultrasonic sensor.
205   DQ1 = constrain(DQf, 1, 100);
206
207   // LDR value mapping.
208   DLl = map(ldrI_val, 0, 1023, 0, 100);
209   DLr = map(ldrD_val, 0, 1023, 0, 100);
210
211   //influence factor.
212   int DLm = (DLl +DLr)/2;
213   int DLsup = DLm + 3;
214   int DLinf = DLm - 3;
215
216   //initialize variables.
217   detR=0; detO=0; detA=0;
218   q1r=0; q2r=0; q3r=0; q1a=0;
219   velRr=140; velRl=140; velAr=200; velAl=200;
220   velr=velRr; vell=velRl;
221
222   if (swarmon == 1){
223       // repulsion.
224       if (DQ1<=rr || DQ2<=rr || DQ3<=rr){
225           if (DQ1<=rr){
226               q1r=1;
227           }
228           if (DQ2<=rr){
229               q2r=1;
230           }
231           if (DQ3<=rr){
232               q3r=1;
233           }
234           //speed and components of the desired direction of repulsion.
235           ddx= -(Q1[0] q1r + Q2[0] q2r + Q3[0] q3r);
236           ddy= -(Q1[1] q1r + Q2[1] q2r + Q3[1] q3r);
237           velr = velRr; vell = velRl; detR=1;

```

```

238     } // end repulsion
239
240     //influence.
241     if (detR==0){
242         if (DLinf<DLl && DLl<DLsup && DLinf<DLr && DLr<DLsup){
243             // attraction.
244             if (ro<DQ1 && DQ1<=ra){
245                 detA=1; velr=velAr; vell=velAl;
246                 ddx=0; ddy=1;
247             } //end attraction.
248         }
249     else{
250         dd2 = 20;
251         if (DLl<DLr){
252             TurnRight(dd2);
253             Stop();
254         }
255     else{
256         TurnLeft(dd2);
257         Stop();
258     }
259 }
260 } //end influence.
261
262 //orientation.
263 if (detR==0 && detA==0){
264     if (DQ1<=ro || DQ2<=ro || DQ3<=ro){
265         detO=1; velr=velRr; vell=velRl;
266         ddx=0; ddy=1;
267     }
268     //out of range.
269     if (detO==0){
270         // if nothing is detected, keep direction.
271         velr = velRr; vell =velRl;
272         ddx=0; ddy=1;
273     } //end out of range.

```

```
274     } //end orientation.
275
276     //calculating the desired direction in radians.
277     float ddrad=atan2(ddy,ddx);
278     dd=ddrad*180/3.14159;
279
280     //conversion to the reference system of the robot.
281     if (dd>-90){
282         dd2=dd-90;
283     }
284     if (dd<=-90){
285         dd2=dd+270;
286     }
287
288     if (dd2>=90 && dd2<180){
289         TurnLeft(90);
290         Forward();
291     }
292     if(dd2<90 && dd2>0){
293         TurnLeft(dd2);
294         Forward();
295     }
296
297     if (dd2<=-90 && dd2>-180){
298         TurnRight(90);
299         Forward();
300     }
301
302     if (dd2>-90 && dd2<0){
303         dd2=abs(dd2);
304         TurnRight(dd2);
305         Forward();
306     }
307
308     if (dd2==0){
309         Forward();
```

```
310     }
311
312     if (dd2>=178 && dd2<=182){
313         if (irI_val>irD_val){
314             TurnRight(90);
315             Forward();
316         }
317         if (irI_val<=irD_val){
318             TurnLeft(90);
319             Forward();
320         }
321     }
322 } // end swarmon
323 else {
324     Stop()
325 }
326 Mostrar(); //function to display data in serial port.
327 velP =(velr+vell)/2;
328 } //End of void loop
329
330 //motion functions.
331 void Forward(){
332     digitalWrite(motorLD, 0);
333     digitalWrite(motorRD, 0);
334     analogWrite(motorLA, vell);
335     analogWrite(motorRA, velr);
336 }
337
338 void Backward(){
339     digitalWrite(motorLD, 1);
340     digitalWrite(motorRD, 1);
341     digitalWrite(motorLA, 0);
342     digitalWrite(motorRA, 0);
343 }
344
345 void TurnRight(int grados){
```

```
346    digitalWrite(motorLD, 0);
347    digitalWrite(motorRD, 1);
348    digitalWrite(motorLA, 1);
349    digitalWrite(motorRA, 0);
350    int time= 3.39*grados;
351    delay(time);
352 }
353
354 void TurnLeft(int grados){
355    digitalWrite(motorLD, 1);
356    digitalWrite(motorRD, 0);
357    digitalWrite(motorLA, 0);
358    digitalWrite(motorRA, 1);
359    int time= 3.39*grados;
360    delay(time);
361 }
362
363 void Stop(){
364    digitalWrite(motorLD, 1);
365    digitalWrite(motorRD, 1);
366    digitalWrite(motorLA, 1);
367    digitalWrite(motorRA, 1);
368 }
369
370 int Ultrasonico(){
371    int distance1=sonar.ping_cm();
372    delay(5);
373    int distance2=sonar.ping_cm();
374    delay(5);
375    int distance3=sonar.ping_cm();
376    int prom=(distance1+distance2+distance3)/3;
377    if (prom==0){
378        prom=100;
379    }
380    return prom;
381 }
```


APÉNDICE C

CÓDIGO EN SCILAB - REGLAS DE COMPORTAMIENTO

```
1  //*****
2  //Estudio de comportamientos emergentes en enjambres de robots basados
3  //factores de influencia
4  //MCIE Mecatronica
5  //UANL - FIME
6  //Erick de Jesus Ordaz Rivas - Mario Aguilera Ruiz
7  //*****
8
9  function [dxdt]=DynamicModel(t,c)
10     // parameters
11     m = .5;      //mass.
12     Ip = 0.005; //inertia moment.
13     d = 0.02;    //distance centroide to axis wheel.
14     r = 0.03;    //radio wheel.
15     R = 0.05;    //distance wheel-center.
16
17     M = [m 0 ; 0 (Ip+m*d^2)];      //inertial matrix.
18     C = [-m*d*c(6)^2 ; m*d*c(5)*c(6)] //matrix of Coriolis.
19     B = [1/r 1/r ; R/r -R/r];      //matriz torque-wheel-movil force.
20
21     ep = [c(5) ; c(4)] - [vd ; c(4)];
```

```

22   ev = [diff(c(5)) ; c(6)] - [0 ; c(6)];
23   ea = [0 ; diff(c(6))];
24
25   Kp = [1 0; 0 1];
26   Kv = [1 0; 0 1];
27   tau= inv(B)*(C + M*(-Kp*ep - Kv*ev + ea));
28
29   dxdt=[cos(c(4)) -d*sin(c(4)); sin(c(4)) d*cos(c(4))]*[c(5); c(6)];
30   c(5);
31   c(6);
32   inv(M)*(B*tau-C)];
33 endfunction
34
35 function [c,t]= Mov(vd,ci) //movil dynamic.
36   ti=0; //initial time.
37   tf=1; //final time.
38   tspan=[ti:0.1:tf]; //time vector.
39   errcatch(98,"pause");
40   c=ode(ci,ti,tspan,DynamicModel);
41   t=tspan;
42 endfunction
43
44 function [res]=SimMov(Tv,rr,ro,ra)
45   Tm= Tv*12; //iterations.
46   lims= Tv*10; //limits.
47   C=zeros(Tv,6); //vehicle states.
48   ri = (lims*0.6); //perceptiveness of influence.
49   velO=2; velR = 0.5; velA = 4; velI = 4; //speed.
50   vd = zeros(Tv,1)+velO; //desired speed.
51   Report=zeros(Tm,Tv,3); //report of results.
52   AlphaR=zeros(Tm,Tv,1); //report of angle alpha.
53   res=zeros(3,5); //results of the center of mass and dispersion.
54   N1=1; N2=(Tm/3); N3=2*(Tm/3); N4=Tm; //time periods.
55
56   figure()
57   plot(lims,lims,'k');
```

```

58     f=get("current_figure");
59     f.background=8;
60
61     //initialize variables.
62     for v=1:Tv
63         C(v,1)=ceil(lims*rand()); //x position.
64         C(v,2)=ceil(lims*rand()); //y position.
65         C(v,3)=0; //movement.
66         C(v,4)=2*%pi*rand(); //orientation.
67         C(v,5)=0; //speed.
68         C(v,6)=0; //angular speed.
69     end
70
71     for m=1:Tm
72         //influence position.
73         if m<Tm then
74             inf =[lims*0.9,lims*0.1];
75         end
76         //value of active and passive influence.
77         if m<N2 | m>N3 then
78             dInf = 0;
79         else
80             dInf = ri;
81         end
82
83         for v=1:Tv
84             [cs,t] = Mov(vd(v),C(v,:)' );
85             [r,c]=size(cs);
86             C(v,:)=cs(:,c)'; //next initial condition, actual condition.
87             vd(v) = velO;
88
89             //this avoids an infinite increment of radians.
90             if C(v,4)>(2*%pi) then
91                 C(v,4)=C(v,4)-(2*%pi);
92             end
93             if C(v,4)<0 then

```

```

94      C(v,4)=C(v,4)+(2*%pi);
95  end
96
97  //limits
98  turnLIM = (%pi/2);
99  if C(v,1)>(lims-(lims*0.05)) then
100      if (0<((C(v,4)*180)/%pi) & ((C(v,4)*180)/%pi)<90) then
101          C(v,4)=C(v,4)+(0.5*rand()+turnLIM);
102      end
103      if (270<((C(v,4)*180)/%pi) & ((C(v,4)*180)/%pi)<360) then
104          C(v,4)=C(v,4)+(0.5*rand()-turnLIM);
105      end
106  end
107  if C(v,2)>(lims-(lims*0.05)) then
108      if (90<((C(v,4)*180)/%pi) & ((C(v,4)*180)/%pi)<180) then
109          C(v,4)=C(v,4)+(0.5*rand()+turnLIM);
110      end
111      if (0<((C(v,4)*180)/%pi) & ((C(v,4)*180)/%pi)<90) then
112          C(v,4)=C(v,4)+(0.5*rand()-turnLIM);
113      end
114  end
115  if C(v,1)<(lims*0.05) then
116      if (180<((C(v,4)*180)/%pi) & ((C(v,4)*180)/%pi)<270) then
117          C(v,4)=C(v,4)+(0.5*rand()+turnLIM);
118      end
119      if (90<((C(v,4)*180)/%pi) & ((C(v,4)*180)/%pi)<180) then
120          C(v,4)=C(v,4)+(0.5*rand()-turnLIM);
121      end
122  end
123  if C(v,2)<(lims*0.05) then
124      if (270<((C(v,4)*180)/%pi) & ((C(v,4)*180)/%pi)<360) then
125          C(v,4)=C(v,4)+(0.5*rand()+turnLIM);
126      end
127      if (180<((C(v,4)*180)/%pi) & ((C(v,4)*180)/%pi)<270) then
128          C(v,4)=C(v,4)+(0.5*rand()-turnLIM);
129      end

```

```

130     end
131
132     //elements detected.
133     DetR=0; DetA=0; DetI=0; DetO=0;
134     ElemR = []; ElemA = [];
135
136     //screening range for repulsion , orientation , attraction and influence.
137     rangR=1.74533; rangA= 0.5235988; rangI=2.61799;
138
139     for z=1:Tv
140         //individual angle with respect to other members of the swarm.
141         angD = atan(C(z,2)-C(v,2),C(z,1)-C(v,1))
142         if angD<0 then
143             angD=angD+(2*%pi);
144         end
145         if angD > (2*%pi) then
146             angD=angD-(2*%pi);
147         end
148
149         //calculation of angles repulsion and attraction.
150         Beta = angD - C(v,4);
151         if Beta < 0 then
152             Beta = Beta + (2*%pi);
153         end
154
155         Gamma = C(v,4) - angD;
156         if Gamma < 0 then
157             Gamma = Gamma + (2*%pi);
158         end
159
160         if Gamma<Beta then
161             Delta = Gamma;
162         else
163             Delta = Beta;
164         end
165

```

```

166      //calculate repulsion distance.
167      if Delta<rangR then
168          if v<>z then //it must not be the same.
169              dr=sqrt((C(v,1)-C(z,1))^2+(C(v,2)-C(z,2))^2);
170          else
171              dr=1000000;
172          end
173      else
174          dr=1000000;
175      end
176
177      //calculate attraction distance.
178      if Delta<rangA then
179          if v<>z then //it must not be the same.
180              da=sqrt((C(v,1)-C(z,1))^2+(C(v,2)-C(z,2))^2);
181          else
182              da=1000000;
183          end
184      else
185          da=1000000;
186      end
187
188      //number of individuals detected in attraction and repulsion radius.
189      if dr<rr then
190          ElemR = [ElemR ; angD];
191          DetR=DetR+1; //a vehicle is detected.
192      end
193
194      if da>ro & da<ra & DetR==0 then
195          ElemA = [ElemA ; angD];
196          DetA=DetA+1; //a vehicle is detected.
197      end
198
199      d=sqrt((C(v,1)-C(z,1))^2+(C(v,2)-C(z,2))^2);
200      if d>rr & d<ro & DetR==0 & DetA ==0 then
201          DetO=DetO+1; //a vehicle is detected.

```

```

202     end
203     end //for of z
204
205     //influence angle
206     angI = atan( inf(2)-C(v,2) , inf(1)-C(v,1) );
207     if angI<0 then
208         angI=angI+(2*%pi);
209     end
210     if angI > (2*%pi) then
211         angI=angI-(2*%pi);
212     end
213
214     //calculating angles repulsion and attraction.
215     BetaI = angI - C(v,4);
216     if BetaI < 0 then
217         BetaI = BetaI + (2*%pi);
218     end
219
220     GammaI = C(v,4) - angI;
221     if GammaI < 0 then
222         GammaI = GammaI + (2*%pi);
223     end
224
225     if GammaI<BetaI then
226         DeltaI = GammaI;
227     else
228         DeltaI = BetaI;
229     end
230
231     //distance of calculated influence.
232     if Delta<rangI then
233         di=sqrt(( inf(1)-C(v,1))^2+( inf(2)-C(v,2))^2);
234     else
235         di = 100000;
236     end
237

```

```

238      //average of detected objects.
239      PromElemR = mean(ElemR)
240      PromElemA = mean(ElemA)
241
242      //repulsion.
243      if DetR > 0 then
244          vd(v) = velR;
245          if PromElemR < %pi then
246              C(v,4)=PromElemR + %pi;
247          else
248              C(v,4)=PromElemR - %pi;
249          end
250      end
251
252      //influence.
253      if (di<dInf & DetR==0) then
254          DetI=1;
255          vd(v) = velI;
256          C(v,4) = (angI*0.6) + (C(v,4)*0.4);
257      end
258
259      //attraction.
260      if (DetA>0 & DetR==0 & DetI==0) then
261          vd(v) = velA;
262          C(v,4)=PromElemA;
263      end
264
265      //orientation.
266      if (DetO>0 & DetR==0 & DetA==0 & DetI==0) then
267          vd(v) = velO;
268          C(v,4) = C(v,4);
269      end
270
271      //out of range
272      if (DetO==0 & DetR==0 & DetA==0 & DetI==0) then
273          vd(v) = velO;

```

```

274         C(v,4) = C(v,4);
275     end
276
277     //calculating degrees for orientation of arrows.
278     phi = C(v,4);
279     angles=(phi*180)/%pi;
280     alpha = C(v,4) - (0.5*%pi);
281
282     if alpha > (2*%pi) then
283         alpha=alpha-(2*%pi);
284     end
285     if alpha < 0 then
286         alpha=alpha+(2*%pi);
287     end
288
289     //Reports.
290     Report(m,v,1) = C(v,1);
291     Report(m,v,2) = C(v,2);
292     Report(m,v,3) = angles;
293     Report(m,v,4) = C(v,5);
294     AlphaR(m,v,1) = alpha;
295 end //for of v
296
297 //Plot vehicles positions.
298 a = gca(); delete(a.children);
299 for v=1:Tv
300     xf(v)=C(v,1)-0.01*sin(AlphaR(m,v,1));
301     yf(v)=C(v,2)+0.01*cos(AlphaR(m,v,1));
302 end
303 nx=[C(:,1)'; xf'];
304 ny=[C(:,2)'; yf'];
305 xarrows(nx,ny,(Tv*3))
306 set(gca(),"auto_clear","off")
307 h_compound = gce();
308
309 if m>N2 & m<N3 then

```

```
310     plot( inf(1), inf(2), 'y.' );
311     h_compound = gce();
312     h_compound.children.mark_size = 10;
313     end
314
315     if m==N1 then
316         pnx1=nx;
317         pny1=ny;
318     end
319     if m==N2-5 then
320         pnx2=nx;
321         pny2=ny;
322     end
323     if m==N3 then
324         pnx3=nx;
325         pny3=ny;
326     end
327     if m==N4 then
328         pnx4=nx;
329         pny4=ny;
330     end
331 end //for of m
332
333 for i=1:4
334     figure()
335     f=get("current_figure"); f.background=8;
336     plot(lims,lims);
337     plot(0,0);
338     xlabel('x-axis')
339     ylabel('y-axis')
340     if i==1 then
341         N=N1;
342         xarrows(pnx1,pny1,(Tv*3));
343         set(gca(),"auto_clear","off");
344     end
345     if i==2 then
```

```

346     N=N2-5;
347     xarrows(pnx2, pny2, (Tv*3));
348     set(gca(), "auto_clear", "off");
349 end
350 if i==3 then
351     N=N3;
352     xarrows(pnx3, pny3, (Tv*3));
353     set(gca(), "auto_clear", "off");
354 end
355 if i==4 then
356     N=N4;
357     xarrows(pnx4, pny4, (Tv*3));
358     set(gca(), "auto_clear", "off");
359 end
360
361 cx = sum(Report(N,:,1))/Tv;
362 cy = sum(Report(N,:,2))/Tv;
363 dy=sqrt(sum((Report(N,:,2)-cy).^2)/Tv);
364 dx=sqrt(sum((Report(N,:,1)-cx).^2)/Tv);
365 area=%pi*dx*dy;
366 res(i,1)=cx;
367 res(i,2)=cy;
368 res(i,3)=dx;
369 res(i,4)=dy;
370 res(i,5)=area;
371 set(gca(), "auto_clear", "off")
372 plot(cx, cy, 'bx')
373 xset("color", 0)
374 xarc(cx-(dx*1.3), cy+(dy*1.3), (dx*1.3)*2, (dy*1.3)*2, 0, 360*64)
375 if i==3 then
376     plot(inf(1), inf(2), 'y. ');
377     h_compound = gce();
378     h_compound.children.mark_size = 10;
379 end
380 end //end for i
381 endfunction

```

```
382
383 function [promres,ress]=simulation(Tv,rr,ro,ra)
384     for i=1:3
385         [res]=SimMov2(Tv,rr,ro,ra)
386         for j=1:4
387             for k=1:5
388                 ress(i,j,k)=res(j,k);
389             end
390         end
391     end
392     for i=1:4
393         for j=1:5
394             promres(i,j)= sum(ress(:,i,j))/3;
395         end
396     end
397 endfunction
```

APÉNDICE D

CÓDIGO EN SCILAB - PROCESAMIENTO DE IMÁGENES

```
1 //*****
2 //Estudio de comportamientos emergentes en enjambres de robots basados
3 //factores de influencia
4 //MCIE Mecatronica
5 //UANL – FIME
6 //Erick de Jesus Ordaz Rivas
7 //*****
8
9 function [res,nr,area,cx,cy,dx,dy]=DetectRobots()
10   RGB = ReadImage(' ');
11   Image = RGB2Gray(RGB);
12   [height width] = size(Image);
13   Image=double(Image);
14   Image=bricontra(Image,brillo,contraste);
15
16   InvertedImage = uint8(255*ones(size(Image,1), size(Image,2))) - Image;
17   Threshold = CalculateOtsuThreshold(InvertedImage); //UMBRAL
18   LogicalImage = SegmentByThreshold(InvertedImage,Threshold);
19
20   StructureElementClose = CreateStructureElement('square',clos)
21   StructureElementClose.Data
```

```
22  StructureElementErode = CreateStructureElement('square',erod)
23  StructureElementErode.Data
24  LogicalImageClose = CloseImage(LogicalImage,StructureElementClose);
25  LogicalImageErode = ErodeImage(LogicalImageClose,StructureElementErode);
26
27  BlobImage = SearchBlobs(LogicalImageErode);
28  FilteredBlobImage = FilterBySize(BlobImage, rang);
29  NumberRobots = double(max(FilteredBlobImage))
30
31  figure();
32  ShowImage(LogicalImageErode, 'LogicalImageErode')
33
34  for h=1:height
35      for w=1:width
36          for nr=1:NumberRobots
37              if (FilteredBlobImage(h,w) == nr) then
38                  rows(h,nr) = h;
39                  columns(w,nr) = w;
40              end
41          end
42      end
43  end
44
45  for nr=1:NumberRobots
46      r(nr) = 0;
47      c(nr) = 0;
48      ar(nr) = 0;
49      ac(nr) = 0;
50      res(nr,2) = 0;
51  end
52
53  Lr = length(rows)/NumberRobots;
54  Lc = length(columns)/NumberRobots;
55
56  for nr=1:NumberRobots
57      r(nr) = sum(rows(:,nr));
```

```

58     for row=1:Lr
59         if rows(row,nr) ~= 0 then
60             ar(nr)=ar(nr)+1;
61         end
62     end
63 end
64
65 for nr=1:NumberRobots
66     c(nr) = sum(columns(:,nr));
67     for col=1:Lc
68         if columns(col,nr) ~= 0 then
69             ac(nr)=ac(nr)+1;
70         end
71     end
72 end
73
74 for nr=1:NumberRobots
75     res(nr,1) = c(nr)/ac(nr);
76     res(nr,2) = h-r(nr)/ar(nr);
77 end
78
79 cx = sum(res(:,1))/nr;
80 cy = sum(res(:,2))/nr;
81 dy=sqrt(sum((res(:,2)-cy).^2)/nr);
82 dx=sqrt(sum((res(:,1)-cx).^2)/nr);
83 area= %pi*dx*dy;
84
85 figure(1);
86 ShowColorImage(RGB, 'Image')
87 set(gca(), "auto_clear", "off")
88 for nr=1:NumberRobots
89     plot(res(nr,1), res(nr,2), 'r..')
90 end
91 plot(cx, cy, 'b.x')
92 xset("color", 0)
93 xarc(cx-(dx*1.3), cy+(dy*1.3), (dx*1.3)*2, (dy*1.3)*2, 0, 360*64)

```

```
94     title('Position_of_the_swarm_members_in_a_virtual_xy-plane');
95     xlabel('x-axis')
96     ylabel('y-axis')
97     set(gca(),'auto_clear','on')
98
99     figure(2);
100     ShowColorImage(ROI, 'Image')
101     set(gca(),'auto_clear','off')
102     for nr=1:NumberRobots
103         plot(res(nr,1),res(nr,2),'r..')
104         for nr2=1:NumberRobots
105             plot([res(nr,1) res(nr2,1)],[res(nr,2) res(nr2,2)],'k')
106         end
107     end
108     plot(cx,cy,'b.x')
109     title('Position_of_the_swarm_members_in_a_virtual_xy-plane');
110     xlabel('x-axis')
111     ylabel('y-axis')
112     set(gca(),'auto_clear','on')
113 endfunction
```


BIBLIOGRAFÍA

- [1] O. Linda and M. Manic. Fuzzy manual control of multi-robot system with built-in swarm behavior. In *Human System Interactions, 2009. HSI '09. 2nd Conference on*, pages 4–9, May 2009.
- [2] E. Ferrante, Ali Emre Turgut, Edgar Duéñez-Guzman, Marco Dorigo, and Tom Wenseleers. Evolution of self-organized task specialization in robot swarms. *PLoS Comput Biol*, 11(8):1–21, 08 2015.
- [3] Ying Tan and Zhong-Yang Zheng. Research advance in swarm robotics. *Defence Technology*, 9(1):18 – 39, 2013.
- [4] M. Aguilera, J. A. Rodríguez-Liñan, and L. M. Torres-Treviño. Change of behavior on swarm robotics using repulsion and attraction. In *2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI)*, pages 169–172, Oct 2015.
- [5] Benjamin T. Fine and Dylan A. Shell. Unifying microscopic flocking motion models for virtual, robotic, and biological flock members. *Autonomous Robots*, 35(2-3):195–219, 2013.
- [6] C. Moeslinger, T. Schmickl, and K. Crailsheim. Emergent flocking with low-end swarm robots. In *Swarm Intelligence*, volume 6234 of *Lecture Notes in Computer Science*, pages 424–431. Springer Berlin Heidelberg, 2010.
- [7] C. Moeslinger, T. Schmickl, and K. Crailsheim. A minimalist flocking algorithm for swarm robots. In *Advances in Artificial Life. Darwin Meets von Neumann*,

- volume 5778 of *Lecture Notes in Computer Science*, pages 375–382. Springer Berlin Heidelberg, 2011.
- [8] A. Brutschy, L. Garattoni, M. Brambilla, G. Francesca, G. Pini, M. Dorigo, and M. Birattari. The tam: abstracting complex tasks in swarm robotics research. *Swarm Intelligence*, 9(1):1–22, 2015.
- [9] Micael S. Couceiro, Patricia A. Vargas, Rui P. Rocha, and Nuno M.F. Ferreira. Benchmark of swarm robotics distributed techniques in a search task. *Robotics and Autonomous Systems*, 62(2):200 – 213, 2014.
- [10] Levent Bayindir. A review of swarm robotics tasks. *Neurocomputing*, 172:292 – 321, 2016.
- [11] F. Ducatelle, A. Di Caro, and M. Gambardella. Cooperative self-organization in a heterogeneous swarm robotic system. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 87–94, New York, NY, USA, 2010. ACM.
- [12] O. Linda and M. Manic. Fuzzy force-feedback augmentation for manual control of multirobot system. *Industrial Electronics, IEEE Transactions on*, 58(8):3213–3220, Aug 2011.
- [13] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.
- [14] F. Ducatelle, Gianni A. Di Caro, C. Pinciroli, and Luca M. Gambardella. Self-organized cooperation between robotic swarms. *Swarm Intelligence*, 5(2):73–96, 2011.
- [15] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, Siggraph'87, pages 25–34, New York, NY, USA, 1987. ACM.

-
- [16] Iain D. Couzin, J. Krause, R. James, Graeme D. Ruxton, and Nigel R. Franks. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218(1):1 – 11, 2002.
- [17] M. Pac, A. Erkmen, and I. Erkmen. Control of robotic swarm behaviors based on smoothed particle hydrodynamics. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 4194–4200, Oct 2007.
- [18] L.C.A. Pimenta, G.A.S. Pereira, N. Michael, R.C. Mesquita, M.M. Bosque, L. Chaimowicz, and V. Kumar. Swarm coordination based on smoothed particle hydrodynamics technique. *Robotics, IEEE Transactions on*, 29(2):383–399, April 2013.
- [19] Z. Kira and M.A. Potter. Exerting human control over decentralized robot swarms. In *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*, pages 566–571, Feb 2009.
- [20] S. Bashyal and G.K. Venayagamoorthy. Human swarm interaction for radiation source search and localization. In *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, pages 1–8, Sept 2008.
- [21] P. Ro. ay, Nnd Newman and S. Srinivasa. *What Types of Interactions Do Bio-Inspired Robot Swarms and Flocks Afford a Human?*, page 504. MIT Press, 2013.
- [22] Shin-Young Jung and M.A. Goodrich. Multi-robot perimeter-shaping through mediator-based swarm control. In *Advanced Robotics (ICAR), 2013 16th International Conference on*, pages 1–6, Nov 2013.
- [23] F. Ducatelle, A. Di Caro, A. Förster, M. Bonani, M. Dorigo, S. Magnenat, F. Mondada, R. O’Grady, C. Pinciroli, P. Rétoznaz, V. Trianni, and M. Gambardella. Cooperative navigation in robotic swarms. *Swarm Intelligence*, 8(1):1–33, 2014.

-
- [24] F. Martinez, E. Jacinto, and D. Acero. Brownian motion as exploration strategy for autonomous swarm robots. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 2375–2380, Dec 2012.
 - [25] M. Pashna, R. Yusof, and R. Rahmani. Oil spill trajectory tracking using swarm intelligence and hybrid fuzzy system. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 1346–1351, July 2014.
 - [26] Guillaume Sartoretti, Max-Olivier Hongler, Marcelo Elias de Oliveira, and Francesco Mondada. Decentralized self-selection of swarm trajectories: from dynamical systems theory to robotic implementation. *Swarm Intelligence*, 8(4):329–351, 2014.
 - [27] Alexandru Bara and Sanda Dale. Dynamic modeling and stabilization of wheeled mobile robot. In *Proceedings of the 5th WSEAS International Conference on Dynamical Systems and Control, CONTROL'09*, pages 87–92, Stevens Point, Wisconsin, USA, 2009. World Scientific and Engineering Academy and Society (WSEAS).
 - [28] P. Zhao, A. Liang, L. Liu, Y. Chen, H. Guan, and X. Yan. An exploration algorithm for a swarm of homogeneous robots. In *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, pages 1–6, Dec 2009.